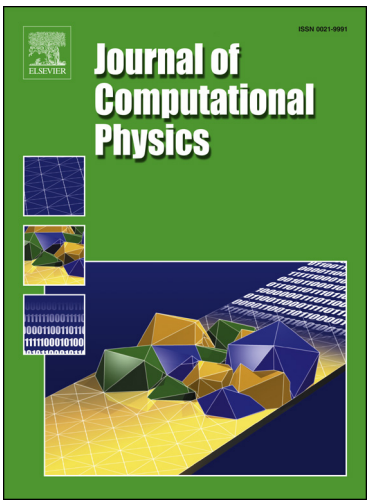
Accepted Manuscript



Higher-order implicit-explicit multi-domain compressible Navier-Stokes solvers

Oscar P. Bruno, Max Cubillos, Edwin Jimenez

PII: S0021-9991(19)30150-0
DOI: https://doi.org/10.1016/j.jcp.2019.02.033
Reference: YJCPH 8532

To appear in: *Journal of Computational Physics*

Received date: 30 May 2018
Revised date: 16 February 2019
Accepted date: 19 February 2019

Please cite this article in press as: O.P. Bruno et al., Higher-order implicit-explicit multi-domain compressible Navier-Stokes solvers, *J. Comput. Phys.* (2019), https://doi.org/10.1016/j.jcp.2019.02.033

Highlights

- Fast FC-based Implicit Navier Stokes solver.
- Spectral algorithm for the Navier-Stokes equations in general domains.
- Multi-domain parallel algorithm.
- Implicit-explicit implementations exhibiting.
- high-order convergence in space and time.

Journal of Computational Physics (2019)

Contents lists available at ScienceDirect

Journal of Computational Physics

journal homepage: www.elsevier.com/locate/jcp

Higher-order implicit-explicit multi-domain compressible Navier-Stokes solvers

Oscar P. Bruno[a,*], Max Cubillos[b], Edwin Jimenez[a]

[a] *Department of Computing and Mathematical Sciences, MC 305-16 California Institute of Technology, Pasadena, CA 91125, USA*
[b] *Riverside Research, Dayton, Ohio, USA*

ARTICLE INFO

Article history:
Received 10 March 2018
Received in final form 17 March 2018
Accepted 30 March 2018
Available online 1 April 2018

Keywords: Navier-Stokes, Alternating direction implicit, Fourier continuation, Overset grid, High-order

ABSTRACT

This paper presents a new class of solvers for the subsonic compressible Navier-Stokes equations in general two- and three-dimensional multi-domains. Building up on the recent single-domain ADI-based high-order Navier-Stokes solvers (Bruno and Cubillos, Journal of Computational Physics 307 (2016) 476-495) this article presents multi-domain implicit-explicit methods of high-order of temporal accuracy. The proposed methodology incorporates: 1) A novel linear-cost implicit solver based on use of high-order backward differentiation formulae (BDF) and an alternating direction implicit approach (ADI); 2) A fast explicit solver; 3) Nearly dispersionless spectral spatial discretizations; and 4) A domain decomposition strategy that negotiates the interactions between the implicit and explicit domains. In particular, the implicit methodology is *quasi-unconditionally stable* (it does not suffer from CFL constraints for adequately resolved flows), and it can deliver *orders of time accuracy between two and six in the presence of general boundary conditions*. As demonstrated via a variety of numerical experiments in two and three dimensions, further, the proposed multi-domain parallel implicit-explicit implementations exhibit high-order convergence in space and time, robust stability properties, limited dispersion, and high parallel efficiency.

1. Introduction

High-order numerical methods for unsteady fluid flow in general domains are subject to computational challenges that arise from the fine spatial discretizations needed to resolve intricate geometric features and complex fluid structures such as boundary layers, vortices, and eddies. Solvers based solely on explicit time advancement can be inefficient due to stringent numerical stability constraints which, in turn, can lead to prohibitively large simulation times even on massively parallel supercomputers. On the other hand, explicit time-marching can be used advantageously in regions where coarse discretizations allow relatively large time steps. In constrast, many classical implicit

*Corresponding author: bruno@acm.caltech.edu

solvers are unconditionally stable and they are therefore exempt from time step restrictions. Nevertheless, implicit methods often require the solution of large systems of equations at each time step, and they can therefore be extremely expensive as well. In this article we propose a hybrid implicit-explicit methodology that, exploiting the strengths of families of explicit and implicit time marching methods, as well as nearly-dispersionless Fourier Continuation (FC) spatial approximations and geometry representation based on use of overset grids, forms the basis for efficient solvers of high-order of accuracy in both space and time.

Modern domain decomposition solution methods for partial differential equations can be traced back to the Schwarz method [1]. Nearly a century later, Volkov presented the first application of the Schwarz method to fully discrete PDE using the method of "composite meshes" [2]. In the decades that followed, similar solution methods based on domain decomposition appeared under various names including composite, overlapping, and overset grid methods. After the introduction of the overset grid method by Volkov and subsequent development by Starius [3], early applications to CFD problems were explored in [4, 5, 6]. Later, various authors [7, 8] demonstrated solvers based on high-order compact finite differences and explicit fourth-order Runge-Kutta time marching as well as an implicit Beam and Warming scheme [9, 10] of *nominal* second-order accuracy (cf. [11]). More recently, the FC methodology was combined with an overset grid approach for the solution of the compressible Navier-Stokes equations in two dimensions [12] and the elasticity equations in three dimensions [13]. A key development in those contributions is the extension of the FC method [14] to overlapping "sub-patch" block-decompositions of larger meshes. Although the contributions [12, 13] have successfully used the overset method in the context of explicit solvers, the goal of extending the framework to implicit and multi-domain implicit-explicit solvers had not been realized until now.

The well-known Beam and Warming method [9, 10], in turn, provides one of the most attractive alternatives to explicit and classical implicit algorithms. Based on the alternating direction implicit method [15] (ADI), the Beam and Warming scheme enables the stable solution of the compressible Navier-Stokes equations in times that grow only linearly with the size of the underlying discretization, and without recourse to either nonlinear iterative solvers or solutions of large linear systems at each time step. However, in practice, previous work in the context of the Beam and Warming method had not demonstrated temporal accuracies beyond first order (cf. [11]). Nevertheless, high-order time accuracy is crucial in long-time simulations or highly-inhomogeneous flows—for which the dispersion inherent in low-order approaches would make it necessary to use inordinately small time-steps.

The aforementioned limitations in the Beam and Warming scheme are addressed in [16, 11]. Based on the backward differentiation formulae (BDF), which are known for their robust stability properties, and the Douglas-Gunn ADI scheme, the BDF-ADI method [11] exhibits *orders of time accuracy between two and six*. Importantly, these algorithms are *quasi-unconditionally stable*—a concept that is defined precisely in [11], and which essentially amounts to true unconditional stability for values of Δt below a certain threshold. Reference [17] presents an extended discussion which places the observed quasi-unconditional stability of the s-order methods, with $2 \leq s \leq 6$, on a sound theoretical basis. As demonstrated in [11], high-order temporal accuracy can be greatly advantageous in cases involving long evolution times or solutions that oscillate rapidly; methods of lower order may be more advantageous under other circumstances.

(It has been suggested in the literature that, owing to their stability characteristics, BDF-based Navier-Stokes solvers of orders higher than two are unsuitable for use in connection with realistic applications. The 2002 reference [18], for example, compares various implicit methods for the Navier-Stokes equations, and it states "Practical experience indicates that large-scale engineering computations are seldom stable if run with BDF4. The BDF3 scheme, with its smaller regions of instability, is often stable but diverges for certain problems and some spatial operators. Thus, a reasonable practitioner might use the BDF2 scheme exclusively for large-scale computations." We note, however, that neither the article [18] nor the references it cites investigate in detail the stability restrictions associated with the BDF methods of order $s > 2$, either theoretically or experimentally. And, importantly, higher-order methods can be useful: as demonstrated in [11] methods of order higher than two give rise to very significant advantages for certain classes of problems—especially for long-time large-scale computations for which the temporal dispersion inherent in low-order approaches would make it necessary to use inordinately small time-steps.)

This paper presents, for the first time, high-order ADI-based Navier-Stokes solvers in a multi-domain implicit-explicit context—thus reaping the benefits of both the explicit and implicit approaches. While the computational cost of the proposed (implicit) BDF-ADI schemes mentioned above grows only linearly with the size of the spatial discretization, these schemes are significantly more expensive per time step than their explicit counterparts—such as the explicit Fourier Continuation solver presented in [12]. Thus the strategy proposed in this work calls for use of multi-domain implicit-explicit solvers—implicit near boundaries where fine discretizations are required to resolve

boundary layers or intricate geometric features (which might require extremely small time steps in an explicit solver), and explicit in regions in which the size of the spatial discretization does not impose significant CFL constraints. (The proposed multi-domain implicit-explicit schemes should not be confused with similarly-named IMEX methods [19] which, e.g., in an advection-diffusion equation incorporate explicit treatment of the convective term and implicit treatment of the diffusive term.)

PDE solvers for structured overset grids generally depend on finite differences (FD) to approximate spatial derivatives, in view of their simplicity, ease of implementation, and limited cost per spatial discretization point. However, achieving high-order spatial accuracy with FD, in practice, is not without challenges. As is well known, a reduction of the dispersion error inherent in FD methods requires an increasing number of points per wavelength for a fixed level of accuracy or use of higher-order methods which typically entail higher costs and restrictive CFL constraints [12, 13, 20]. In addition, the use of high-order biased stencils near and at domain boundaries can lead to numerical instability. Although compact schemes [21, 22] and summation-by-parts (SBP) operators [23, 24, 25, 26] are effective in restoring stability to FD methods, they often do so at the expense of reduced accuracy orders near the boundary. Certain techniques exist which offer some improvements in regards to dispersion error, such as dispersion-relation-preserving schemes [27, 28] and modified Padé operators [21]. Although these methods can be tuned to provide perfect dispersion for a few select frequencies, there is a corresponding reduction in the order of accuracy of the FD scheme, and these methods also do not address the broadband dispersion inherent in nonlinear problems.

Spectral methods, on the other hand, are an attractive alternative to tackle these challenges [29, 30, 31]: these methods generally require fewer discretization points for a given accuracy tolerance compared to finite differences, and they reproduce the dispersion characteristics of the PDE remarkably well (perfectly in the case of Fourier methods). Unfortunately, polynomial spectral methods require clustering of points at the boundaries of the domain, resulting in severe time step restrictions for explicit methods. Classical Fourier methods use uniformly-spaced points but they are only applicable to periodic problems—otherwise they suffer from the Gibbs phenomenon and the order of accuracy deteriorates to first order in the interior of the domain (see, e.g., [29, Ch. 2.2]).

The recently introduced Fourier Continuation method (FC), which is central to our solver methodology, provides spectral-like resolution in non-periodic contexts without requiring fine meshes. The FC method produces an interpolating Fourier series representation by relying on a "periodic extension" of a given function that closely approximates it in its original domain but which is periodic on a slightly enlarged domain. In the context of explicit algorithms, following [12, 13, 20] the FC spatial discretizations are used in conjunction with the Adams-Bashforth (AB) method [32, Ch. 3.9] of orders two through four. As shown in previous references [12, 13, 20, 14], the resulting FC time-domain solvers (whether explicit or implicit) give rise to significantly improved dispersion properties, low computing costs, high accuracies and favorable spectral asymptotics in CFL constraints—as well as parallelization with near-perfect scaling. In particular, the explicit solver is significantly more accurate than other explicit methods for similar computing times, and significantly faster than other schemes for a given accuracy; cf. [12].

Unlike previous general Navier-Stokes solvers, all of the methods presented in this article, including the explicit, implicit, and multi-domain solvers mentioned above, enjoy *near spatial dispersionlessness* as well as *higher orders of accuracy in both space and time*. Such desirable characteristics are demonstrated, in particular, by means of implicit and explicit solutions in single domains as well as multi-domain implicit-explicit solutions *with non-trivial boundary conditions*—including no-slip boundary conditions at walls, and, depending on the case under consideration, absorbing boundary conditions and inflow conditions. The proposed BDF-ADI solvers, further, enjoy the properties of quasi-unconditional stability, dispersionlessness, and high-order accuracy in time. The multi-domain implicit-explicit solver, in turn, is demonstrated, in particular, with results of two-dimensional flow past a cylinder and three-dimensional flow past a sphere. These are the first high-order time-accurate solutions produced by means of a multi-domain Navier-Stokes solver with an implicit component.

2. Governing equations

Our proposed algorithms are used to solve the Navier-Stokes equations for viscous, compressible flow, written in non-conservative form, for the velocity $\mathbf{u}$, temperature T and density ρ in a perfect gas in a d-dimensional domain $\Omega \subset \mathbb{R}^d$ ($d = 2$ or $d = 3$). We assume that the pressure p, density ρ, temperature T satisfy the ideal gas law $p = \rho RT$, with gas constant R. The heat flux q and the temperature gradient ∇T, in turn, are related by the isotropic Fourier law $q = -\kappa \nabla T$, where κ is the temperature-dependent thermal conductivity. Using characteristic values L_0, u_0, ρ_0, T_0, μ_0 and κ_0 for length, velocity, density, temperature, viscosity and heat conductivity, respectively, the Navier-Stokes

equations can be written in the non-dimensional form [33]

$$\rho_t + \nabla\cdot(\rho\mathbf{u}) = 0 \tag{1a}$$

$$\mathbf{u}_t + \mathbf{u}\cdot\nabla\mathbf{u} + \frac{1}{\gamma \mathrm{Ma}^2}\frac{1}{\rho}\nabla(\rho T) = \frac{1}{\mathrm{Re}}\frac{1}{\rho}\nabla\cdot\boldsymbol{\sigma} \tag{1b}$$

$$T_t + \mathbf{u}\cdot\nabla T + (\gamma-1)T\nabla\cdot\mathbf{u} = \frac{\gamma}{\mathrm{RePr}}\frac{1}{\rho}\nabla\cdot(\kappa\nabla T) + \frac{\gamma(\gamma-1)\mathrm{Ma}^2}{\mathrm{Re}}\frac{1}{\rho}\Phi \tag{1c}$$

($\mathbf{u} = (u_i)$, $\boldsymbol{\sigma} = (\sigma_{ij})$, $i, j = 1, \ldots, d$) where $\gamma = c_p/c_v$ is the ratio of specific heats, $\mathrm{Re} = \rho_0 u_0 L_0/\mu_0$ is the Reynolds number and $\mathrm{Ma} = u_0/\sqrt{\gamma R T_0}$ is the Mach number (with gas constant $R = c_p - c_v$), and where $\mathrm{Pr} = \mu_0 c_p/\kappa_0$ is the Prandtl number. The non-dimensional primitive variables in these equations are the velocity vector $\mathbf{u}$, the density ρ and the temperature T; the quantities $\boldsymbol{\sigma}$ and Φ, in turn, denote the Newtonian deviatoric stress tensor and the viscous dissipation function, respectively: letting $\mathbf{I}$ denote the identity tensor, we have

$$\boldsymbol{\sigma} = \mu\left(\nabla\mathbf{u} + \nabla\mathbf{u}^{\mathrm{T}} - \frac{2}{3}(\nabla\cdot\mathbf{u})\mathbf{I}\right) \quad \text{and} \quad \Phi = \sum_{ij}\sigma_{ij}\partial_{x_i}u_j.$$

We assume that μ and κ are functions of temperature alone and, using Sutherland's law [33, pp. 28–30], we express them as

$$\kappa = \frac{1+S_\kappa}{T+S_\kappa}T^{3/2} \quad \text{and} \quad \mu = \frac{1+S_\mu}{T+S_\mu}T^{3/2}, \tag{2}$$

where S_κ and S_μ are the non-dimensionalized Sutherland constants. The governing system (1) is completed by imposing the appropriate boundary conditions for a given configuration; see e.g. [33, Sec. 1-4]. A detailed description of the boundary conditions for each example is given in the numerical results section.

3. Fourier continuation spatial approximation

The Fourier continuation (FC) method extends the advantages of the classical Fourier expansion (dispersionlessness and high-order accuracy, in particular) to general non-periodic domains and it is the foundation of the spatial approximation used in our multi-domain solver. The Fourier continuation (FC) method produces an interpolating Fourier series by constructing a "periodic extension" of a given function that closely approximates it in its original domain but which is periodic over a slightly larger domain. In other words, given a function f defined, without loss of generality, on the unit interval $f : [0,1] \to \mathbb{R}$, the FC method produces a periodic function $f^c : [0, L] \to \mathbb{R}$ defined on an extended interval, with $L > 1$, which closely approximates f on the original interval $[0, 1]$.

Fully discrete Fourier continuation algorithms generally proceed as follows: given N equispaced discretization points $\{x_i\}_{i=0}^{N-1} \subset [0,1]$, $x_i = ih$, $h = 1/(N-1)$, together with the function values $\{f_i\}_{i=0}^{N-1}$, $f_i = f(x_i)$, the FC method produces a L-periodic trigonometric polynomial

$$f^c(x) = \sum_{k=-M}^{M} a_k e^{\frac{2\pi i k}{L}x}, \tag{3}$$

that matches the given values of f at x_i, $i = 0, \ldots, N-1$. After forming the FC interpolant (3), derivatives of the function can be computed easily by termwise differentiation:

$$\frac{\partial f}{\partial x} \approx \frac{\partial f^c}{\partial x} = \sum_{k=-M}^{M} a_k \frac{2\pi i k}{L} e^{\frac{2\pi i k}{L}x}.$$

Here $M = (N + C - 1)/2$, where C is the number of additional continuation points produced by the FC algorithm (Gram or SVD) described in the next two subsections; without loss of generality, we assume that $N + C$ is an odd positive integer.

3.1. Fourier continuation: FC(SVD)

The FC algorithm presented in [34] for high-order surface representations computes the coefficients a_k of the series (3) as the solution of a least-squares minimization problem

$$\{a_k\} = \arg\min_{a_k} \sum_{i=0}^{N-1} |f_i - f^c(x_i)|^2,$$

which is found in practice by means of the singular value decomposition (SVD). This FC(SVD) method is, unfortunately, far too expensive for time-dependent problems. To circumvent this difficulty, an accelerated method was developed in [14] which allows for Fourier continuation of functions on the basis of a small number $d = d_\ell, d_r$ of points at the left and right ends of the interval and a projection onto a Gram polynomial basis whose FC extensions are precomputed via a high-precision SVD. In effect, this procedure produces a "basis" of continuation functions that can be utilized by a PDE solver. The following section presents a high-level description of the "FC(Gram)" method; see [12, 13] for more detailed discussions in these regards.

3.2. Accelerated Fourier continuation: FC(Gram)

Let $\mathbf{f} = (f_0, \ldots, f_{N-1})^T$ be the column vector containing N values of a given smooth function f in the domain $[0, 1]$ at the equispaced points $x_i = ih$, $0 \le i \le N-1$, and let $\mathbf{f}^c = (f^c_0, \ldots, f^c_{N+C-1})^T$ be the vector of $N + C$ continuation values ($C > 0$) on the extended domain $[0, L]$ at the points $x_i = ih$, $0 \le i \le N + C - 1$, the first N of which coincide with the points of the original interval. For notational simplicity, we also use the periodic continuation of this discrete function of x_i to the whole line by defining

$$x_{i+m(N+C)} = (i + m(N + C))h \quad \text{and} \quad f^c_{i+m(N+C)} \equiv f^c_i \quad \text{for all} \quad m \in \mathbb{Z}. \tag{4}$$

The vector $\mathbf{f}^c$ satisfies a matrix equation of the form [12]

$$\mathbf{f}^c = \begin{pmatrix} I \\ A \end{pmatrix} \mathbf{f}, \tag{5}$$

where I is the $N \times N$ identity matrix and where A is a certain $C \times N$ matrix. The FC(Gram) algorithm produces the C continuation function values using only a small number of points d_ℓ and d_r at the left and right ends of the interval respectively. That is, the matrix A in equation (5) can be expressed in the form

$$A\mathbf{f} = A_\ell \begin{pmatrix} f_0 \\ \vdots \\ f_{d_\ell - 1} \end{pmatrix} + A_r \begin{pmatrix} f_{N-d_r} \\ \vdots \\ f_{N-1} \end{pmatrix},$$

where A_ℓ and A_r are respectively $C \times d_\ell$ and $C \times d_r$ matrices defining a smooth "blend to zero" operation—i.e., A_ℓ takes the function values $f_0, \ldots, f_{d_\ell - 1}$ at $x_0, \ldots, x_{d_\ell - 1}$ and provides a smooth continuation to the left terminating with the function values 0 at the points $x_{-C-d_r}, \ldots, x_{-C-1}$. Similarly, A_r provides a continuation to the right with values $f_{N-d_r}, \ldots, f_{N-1}$ at $x_{N-d_r}, \ldots, x_{N-1}$ and 0 at the points $x_{N+C}, \ldots, x_{N+C+d_\ell - 1}$.

More precisely, the blend-to-zero operations proceed as follows: Without loss of generality let $d_\ell = d_r = d$ (all FC computations in this article satisfy this assumption) and define $\delta = 1 - x_{N-d}$ to be the width of the left and right fringe regions. The rightward extension is obtained by applying the FC(SVD) algorithm described in the previous section to a certain polynomial $p(x)$ defined on $[1 - \delta, L + \delta]$ and extended periodically to the interval $[1 - \delta, L + \delta + (L - 1)]$. The polynomial $p(x)$ is the interpolant of the data $f_{N-d}, \ldots, f_{N-1}$ at the points $x_{N-d}, \ldots, x_{N-1}$ and zero at the points $x_{N+C}, \ldots, x_{N+C+d}$. The resulting Fourier series is then sampled at the continuation points $x_N, \ldots, x_{N+C-1}$ to obtain the function values $p^c_N, \ldots, p^c_{N+C-1}$. Similarly, the leftward extension is obtained by applying the same procedure to the polynomial $q(x)$ interpolating zero at the points $x_{N-d}, \ldots, x_{N-1}$ and the data $\{f_0, \ldots, f_{d-1}\}$ at the points $x_{N+C}, \ldots, x_{N+C+d}$. Once the left and right continuations are computed, the values of the function f^c are simply the sum of the left and right extension values: $f^c_i = p^c_i + q^c_i$, $N \le i \le N + C - 1$. The entire procedure is illustrated in Figure 1.

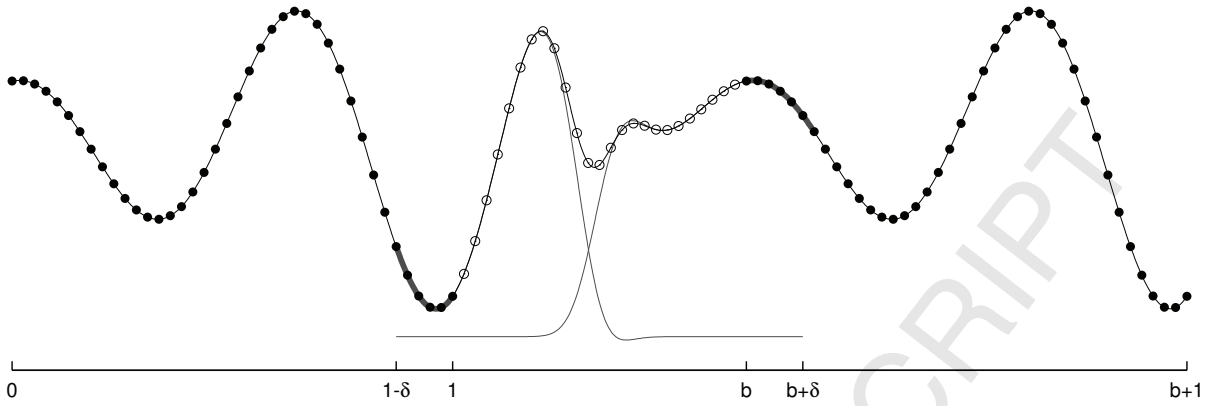


Fig. 1. Illustration of the FC(Gram) method, showing the original function values on the b-periodic domain (solid circles) together with the continuation values (open circles) which are obtained by summing the left and right blend-to-zero extensions (thin gray lines). The thick black curves indicate the polynomial approximations in the fringe regions which are used to produce the blend-to-zero extensions.

Computing the SVDs necessary to complete the continuation procedure described above can be done once for all time on appropriate bases of Gram polynomials in a precomputation stage. The continuation operation (5) is then divided into two steps, which amounts to decomposing the matrices A_ℓ and A_r into the products

$$A_\ell = B_\ell Q_\ell, \quad A_r = B_r Q_r.$$

The $d \times d$ matrices Q_ℓ and Q_r are projections onto orthogonal bases of Gram polynomials which are zero at the right and left fringe points, respectively; and the $C \times d$ matrices B_ℓ and B_r are, respectively, the precomputed left and right blend-to-zero extensions on those bases, as described in the previous paragraph.

Remark 1. *For all numerical examples presented in this article using the FC spatial approximation, the number of left and right fringe points is $d = 5$ and the number of continuation points is $C = 25$. For simplicity, the biased order extensions introduced in [12] are not used.*

Remark 2. *We emphasize that, for a given block in a multi-block mesh, the spatial derivatives along a given direction in parameter space are computed one dimension at a time using the one-dimensional FC(Gram) algorithm. Note that the periodic extension* (5) *used to compute the discrete Fourier coefficients of the FC interpolant* (3) *is an intermediate step that does not alter the underlying domain in any way. Once the FC interpolant is formed, it is differentiated term-by-term to obtain the desired derivative approximation. In the context of the multidomain implicit-explicit solver proposed in this paper, the one-dimensional FC algorithm is applied to all implicit and explicit domains.*

Remark 3. *As expected, the assumed smoothness of the approximated function f is required for high-order convergence in the FC method. As a result, unless special methodologies are used to deal with geometric singularities, FC-based PDE solvers only yield high accuracy when applied to PDEs on smooth domains [14, 35].*

4. Time marching

For clarity, it will be convenient to express the governing equations in terms of a $(d + 2)$-dimensional solution vector $Q = (\mathbf{u}^{\mathrm{T}}, T, \rho)^{\mathrm{T}}$ so that the system (1) becomes

$$Q_t = \mathcal{P}(Q, x, t), \quad x \in \Omega, \quad t \geq 0, \tag{6}$$

where $\mathcal{P}$ is a vector-valued nonlinear differential operator. Note that the x and t dependence of the operator $\mathcal{P}$ allows for source terms which depend on both space and time.

4.1. Explicit time marching

Following [12], an Adams-Bashforth (AB) explicit time marching method [32, Ch. 3.9] is employed in all subdomains where relatively large spatial discretizations do not give rise to stringent CFL constraints. Although the fourth order explicit Runge-Kutta (RK4) scheme is a popular choice for Navier-Stokes time-stepping, the specification of high-order boundary conditions at intermediate stages requires special treatment [36, 37]. Further, both the AB and BDF methods used in our hybrid methodology require enforcement of boundary conditions at the same time level—which leads to a simpler algorithm than would result from use of the multi-stage RK4 method. For convection-dominated problems, however, use of the RK4 scheme might prove a valuable alternative to AB4—since the stability domain of the former method makes a larger intersection with the imaginary axis than does the stability domain of the latter, so that, with a similar cost, a stability-limited time step consisting of four stages of RK4 may be somewhat longer than one stability-limited time step of AB4.

In this article, AB methods of orders 2 to 4 are used, depending on the desired order of time accuracy. Given the form of the PDE (6), the AB scheme of order s for the approximate solution Q^{n+1} at time $t = t^{n+1}$ is given by

$$Q^{n+1} = Q^n + \Delta t \sum_{j=0}^{s-1} b_j \mathcal{P}(Q^{n-j}, x, t^{n-j}); \tag{7}$$

the AB coefficients b_j are summarized in Table 1. Explicit domains in all the numerical examples in this article use

Table 1. Coefficients for AB methods of orders $s = 1, \ldots, 4$.

s	b_0	b_1	b_2	b_3
1	1			
2	$\frac{3}{2}$	$-\frac{1}{2}$		
3	$\frac{23}{12}$	$-\frac{4}{3}$	$\frac{5}{12}$	
4	$\frac{55}{24}$	$-\frac{59}{24}$	$\frac{37}{24}$	$-\frac{3}{8}$

Dirichlet-type boundary conditions. Enforcement is accomplished by injecting the boundary values at all boundary points at the end of each time step, as described in [12].

4.2. Implicit time marching

In subdomains where a fine spatial resolution is required, such as in the vicinity of boundary layers or near complex geometric regions, we use an implicit time marching method that is unconditionally stable for "adequately-resolved" flows as explained in [11]. A detailed description of our BDF-based ADI time marching schemes is given in [11] but, for completeness, a brief summary is provided in what follows.

To derive our BDF-based ADI time marching schemes we consider the following quasilinear-like curvilinear form [11] of the Navier-Stokes equations (1) for the solution vector $Q = Q(\xi, \eta, \zeta, t)$,

$$\begin{aligned} &Q_t + M^{\xi,1}(Q)\frac{\partial}{\partial \xi}Q + M^{\eta,1}(Q)\frac{\partial}{\partial \eta}Q + M^{\zeta,1}(Q)\frac{\partial}{\partial \zeta}Q \\ &+ M^{\xi,2}(Q)\frac{\partial^2}{\partial \xi^2}Q + M^{\eta,2}(Q)\frac{\partial^2}{\partial \eta^2}Q + M^{\zeta,2}(Q)\frac{\partial^2}{\partial \zeta^2}Q \\ &+ M^{\xi\eta}(Q)\frac{\partial^2}{\partial \xi \partial \eta}Q + M^{\xi\zeta}(Q)\frac{\partial^2}{\partial \xi \partial \zeta}Q + M^{\eta\zeta}(Q)\frac{\partial^2}{\partial \eta \partial \zeta}Q + M^0(Q)Q = 0. \end{aligned} \tag{8}$$

where the various M matrices ($M^{\xi,1}$, $M^{\xi,2}$, etc.) are matrix-valued functions of Q. Explicit expressions for the M matrices in equation (8) are given in [11]. The physical domain $\Omega \subset \mathbb{R}^3$, whose coordinates are (x, y, z), is assumed to be the image of smooth invertible mappings $x = x(\xi, \eta, \zeta)$, $y = y(\xi, \eta, \zeta)$, $z = z(\xi, \eta, \zeta)$ of a computational domain (which we take to be a cube $C = [\ell_1, \ell_2]^3$ for $\ell_1, \ell_2 \in \mathbb{R}$) with coordinates (ξ, η, ζ).

Throughout this section we consider a semi-discrete form of (6) where only the time variable is discretized and spatial operators are assumed to be continuous. Let Q^j denote the numerical approximation of Q at time $t = t^j$. Then, the well-known implicit $(\Delta t)^{s+1}$ locally-accurate ($(\Delta t)^s$ globally-accurate) order-s BDF formula applied to (6) yields

$$Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} + b\Delta t\, \mathcal{P}(Q^{n+1}, x, t^{n+1}), \tag{9}$$

where a_k and b are the BDF coefficients of order s. Table 2 shows the BDF coefficients for orders $s = 1$ through $s = 6$. The description of our ADI methodology is simplified using the following variable-coefficient differential operators

Table 2. Coefficients for BDF methods of orders s with $s = 1, \ldots, 6$.

s	a_0	a_1	a_2	a_3	a_4	a_5	b
1	1						1
2	$\frac{4}{3}$	$-\frac{1}{3}$					$\frac{2}{3}$
3	$\frac{18}{11}$	$-\frac{9}{11}$	$\frac{2}{11}$				$\frac{6}{11}$
4	$\frac{48}{25}$	$-\frac{36}{25}$	$\frac{16}{25}$	$-\frac{3}{25}$			$\frac{12}{25}$
5	$\frac{300}{137}$	$-\frac{300}{137}$	$\frac{200}{137}$	$-\frac{75}{137}$	$\frac{12}{137}$		$\frac{60}{137}$
6	$\frac{360}{147}$	$-\frac{450}{147}$	$\frac{400}{147}$	$-\frac{225}{147}$	$\frac{72}{147}$	$-\frac{10}{147}$	$\frac{60}{147}$

$$\mathcal{A}[R] = \sum_{j=0}^{2} M^{\xi,j}(R)\frac{\partial^j}{\partial \xi^j} \tag{10a}$$

$$\mathcal{B}[R] = \sum_{j=1}^{2} M^{\eta,j}(R)\frac{\partial^j}{\partial \eta^j} \tag{10b}$$

$$C[R] = \sum_{j=1}^{2} M^{\zeta,j}(R)\frac{\partial^j}{\partial \zeta^j} \tag{10c}$$

$$\mathcal{G}[R] = M^{\xi\eta}(R)\frac{\partial^2}{\partial \xi \partial \eta} + M^{\xi\zeta}(R)\frac{\partial^2}{\partial \xi \partial \zeta} + M^{\eta\zeta}(R)\frac{\partial^2}{\partial \eta \partial \zeta}, \tag{10d}$$

where R is a given $(d + 2)$-dimensional vector valued function. Note that we define $M^{\xi,0}(R) \equiv M^0(R)$. For example, if we apply the differential operator $\mathcal{A}[R]$ to a vector function S, we obtain

$$\mathcal{A}[R]S = M^{\xi,0}(R)S + M^{\xi,1}(R)\frac{\partial S}{\partial \xi} + M^{\xi,2}(R)\frac{\partial^2 S}{\partial \xi^2} \tag{11}$$

(and similarly for $\mathcal{B}$, C, and $\mathcal{G}$). Now equation (9) can be rewritten as

$$\left(I + b\Delta t\, \mathcal{A}\left[Q^{n+1}\right] + b\Delta t\, \mathcal{B}\left[Q^{n+1}\right] + b\Delta t\, C\left[Q^{n+1}\right]\right) Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\, \mathcal{G}\left[Q^{n+1}\right] Q^{n+1}. \tag{12}$$

Previous ADI-based Navier-Stokes solvers have relied on either linearization or iterations to adequately account for nonlinear terms. In contrast, the methods proposed in this paper employ polynomial extrapolations of order p

$$\widetilde{Q}_p^{n+1} \equiv \sum_{k=0}^{p-1} (-1)^k \binom{p}{k+1} Q^{n-k} \quad (p \geq 1) \tag{13}$$

(with $p = s$) to approximate the matrix-valued functions M in the operators (10). The extrapolation order is selected so that high-order-accurate approximations of the nonlinear terms at time t^{n+1} are obtained.

Using an extrapolated solution, we obtain a variable-coefficient but *linear* approximation to (12)

$$(I + b\Delta t\,\mathcal{A}_s + b\Delta t\,\mathcal{B}_s + b\Delta t\,C_s)\,Q^{n+1} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{G}_s Q^{n+1} \tag{14}$$

for Q^{n+1}, where we have defined the operators

$$\mathcal{A}_s = \mathcal{A}[\widetilde{Q}_s^{n+1}], \quad \mathcal{B}_s = \mathcal{B}[\widetilde{Q}_s^{n+1}], \quad C_s = C[\widetilde{Q}_s^{n+1}], \quad \mathcal{G}_s = \mathcal{G}[\widetilde{Q}_s^{n+1}]. \tag{15}$$

Clearly, if we use an order s extrapolation for Q^{n+1} for the operators (15), the equations (14) are equivalent to the corresponding $(s+1)$-th order equation (12) up to an error of order $(\Delta t)^{s+1}$, and therefore they themselves are locally accurate to order $(s+1)$ in time.

As shown in [11], it is possible to factor equations (14) in a form that minimizes the work needed to form the right-hand sides of the ADI scheme which, at the same, simplifies the enforcement of boundary conditions. The Douglas-Gunn splitting form that we use is

$$(I + b\Delta t\,\mathcal{A}_s)\,Q^{*} = \sum_{k=0}^{s-1} a_k Q^{n-k} - b\Delta t\,\mathcal{G}_s \widetilde{Q}_s^{n+1} - b\Delta t\,(\mathcal{B}_s + C_s)\,\widetilde{Q}_{s-1}^{n+1} \tag{16a}$$

$$(I + b\Delta t\,\mathcal{B}_s)\,Q^{**} = Q^{*} + b\Delta t\,\mathcal{B}_s \widetilde{Q}_{s-1}^{n+1} \tag{16b}$$

$$(I + b\Delta t\,C_s)\,Q^{n+1} = Q^{**} + b\Delta t\,C_s \widetilde{Q}_{s-1}^{n+1}. \tag{16c}$$

A remarkable property of the Douglas-Gunn splitting (16) is that applying the same boundary conditions for Q at time $t = t^{n+1}$ to Q^* and Q^{**} preserves the overall $(\Delta t)^{s+1}$ truncation error (see [11]). Thus, to advance the solution from time step t^n to t^{n+1}, we can impose Dirichlet-type boundary conditions for the intermediate-time unknowns $Q^* = (\mathbf{u}^{*\mathrm{T}}, T^*, \rho^*)^{\mathrm{T}}$ and $Q^{**} = (\mathbf{u}^{**\mathrm{T}}, T^{**}, \rho^{**})^{\mathrm{T}}$ as follows

$$\begin{pmatrix} \mathbf{u}^*(\xi,\eta,\zeta) \\ T^*(\xi,\eta,\zeta) \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,\zeta,t^{n+1}) \\ g^{T}(\xi,\eta,\zeta,t^{n+1}) \end{pmatrix}, \quad \text{for } \xi = \ell_1, \ell_2 \text{ and } \eta,\zeta \in [\ell_1,\ell_2] \tag{17a}$$

$$\begin{pmatrix} \mathbf{u}^{**}(\xi,\eta,\zeta) \\ T^{**}(\xi,\eta,\zeta) \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,\zeta,t^{n+1}) \\ g^{T}(\xi,\eta,\zeta,t^{n+1}) \end{pmatrix}, \quad \text{for } \eta = \ell_1, \ell_2 \text{ and } \xi,\zeta \in [\ell_1,\ell_2] \tag{17b}$$

$$\begin{pmatrix} \mathbf{u}^{n+1}(\xi,\eta,\zeta) \\ T^{n+1}(\xi,\eta,\zeta) \end{pmatrix} = \begin{pmatrix} g^{\mathbf{u}}(\xi,\eta,\zeta,t^{n+1}) \\ g^{T}(\xi,\eta,\zeta,t^{n+1}) \end{pmatrix}, \quad \text{for } \zeta = \ell_1, \ell_2 \text{ and } \xi,\eta \in [\ell_1,\ell_2]. \tag{17c}$$

Note that the results in Section 10 demonstrate that the expected order of accuracy is also achieved in the case of general boundary conditions, including cases in which time-dependent, inflow, and absorbing boundary conditions are specified.

We remark that for numerical stability, a mild Padé-type sixth-order filter [38] is used with both explicit and implicit time marching.

4.2.1. Stability of the BDF-ADI algorithm

In this section we review the stability properties of the BDF-ADI methods. In [17] it was proved that the second order BDF-ADI method is unconditionally stable for linear constant coefficient advection and parabolic equations including mixed derivative terms (treated via the algorithm described in Section 4.2 above) in two dimensional space using either Fourier spectral collocation with periodic boundary conditions or Legendre collocation in a square with homogeneous Dirichlet boundary conditions. In particular, it was shown that temporal extrapolations used for the splitting and mixed derivative terms in the BDF-ADI method do not affect the unconditional stability enjoyed by the fully implicit BDF method.

The previous contributions, including [17], do not present stability proofs for the BDF-ADI methods of order higher than 2, but, in order to provide insights into the stability properties arising from the BDF time-stepping scheme in the context of time-domain PDE solvers, [17, Sec. 5.1] investigates the stability of the BDF schemes of order $s \geq 2$ under periodic boundary conditions and Fourier discretizations. In particular, it is shown in that section that the BDF schemes for the advection-diffusion equation

$$u_t + \alpha \cdot \nabla u = \beta \Delta u \tag{18}$$

are quasi-unconditionally stable. Here u denotes an unknown function in $d = 1, 2$, and 3 spatial dimensions with constant coefficients $\alpha \in \mathbb{R}^d$ and $\beta > 0$. The precise definition of quasi-unconditional stability is as follows:

Definition 1. *Let Ω_h be a family of spatial discretizations of a domain Ω controlled by a mesh-size parameter h and let Δt be a temporal step size. A numerical method for the solution of the PDE $Q_t = \mathcal{P}\, Q$ in Ω is said to be* **quasi-unconditionally stable** *if there exist positive constants M_h and M_t such that for each $\Delta t < M_t$ the method is stable for arbitrarily small values of h—provided $h < M_h$.*

The following theorem is established in [17].

Theorem 1. *The Fourier-based BDF scheme of order s for the problem* (18) *with $3 \le s \le 6$ is quasi-unconditionally stable with constants $M_t = \frac{\beta}{\|\alpha\|^2} m_C$ and $M_h = \infty$. Approximate values of the s-dependent constant m_C are given in Table 3.*

Table 3. Numerical values of the constant m_C appearing in Theorem 1 incorporating, additionally, the full unconditional stability result [17, Sec.] for $s = 2$.

s	2	3	4	5	6
m_C	∞	14.0	5.12	1.93	0.191

In other words, although BDF schemes of order greater than two are not A-stable, the periodic, Fourier-based BDF algorithms for the advection-diffusion equation do not suffer from CFL-like restrictions for any temporal time-step satisfying $\Delta t < \frac{\beta}{\|\alpha\|^2} m_C$, where m_C is the Δx-independent constant listed in Table 3 for the various orders of temporal accuracy. In particular, for any fixed $\Delta t < \frac{\beta}{\|\alpha\|^2} m_C$, the s-order methods are stable for arbitrarily small spatial discretizations ($2 \le s \le 6$).

It is worth pointing out that the theorem applies, in particular, to BDF schemes of order 3 and 4, whose stability regions do not include a segment of the imaginary axis in a neighborhood of the origin [39, Table 2.2] (unlike BDF orders 1, 2, 5, and 6). One might expect that this property could be problematic for small viscosity, but Table 3 shows that BDF orders 3 and 4 are *more* stable than BDF orders 5 and 6 for sufficiently fine spatial discretizations. Although quasi-unconditional stability was not established for the BDF-ADI method, numerical illustrations in Section 5.2 of [17] demonstrate quantitatively, via a numerical study, that the quasi-unconditional stability property carries over to the Navier-Stokes equations as well.

5. Dispersion properties of FC-based solvers

In this section we present computational results that demonstrate the favorable numerical dispersion properties of FC-based solvers.

Consider the advection equation over the interval $[x_l, x_r] = [0, 100]$

$$\begin{cases} u_t + u_x = 0, & (x,t) \in [x_l, x_r] \times (0, T], \\ u(x,0) = u_0(x), & x \in [x_l, x_r]. \end{cases} \tag{19}$$

The initial condition $u_0(x)$ and boundary condition at $x = x_l$ are chosen so that the exact solution u is a wave packet given by

$$u(x,t) = A_0 \exp\left[-\ln 2 \left(\frac{x - x_0 - t}{w} \right)^2 \right] \sin\left[\kappa\,(x - x_0 - t)\right], \tag{20}$$

where the amplitude $A_0 = 1.0$, half-width $w = 1.0$, source center $x_0 = 0$, and wavenumber $\kappa = 10$. We solve equation (19) up to a final time $T = 80$ (when the wave packet has travelled more than 127 wavelengths). To assess the impact of our overset grid methodology on numerical dispersion, we solve (19) over a single-domain grid and also on a 10-sub-domain overlapping grid. The single-domain is represented by a uniformly-spaced grid defined by

$$\mathcal{I} = \left\{ x_j \;:\; x_j = x_l + (j-1)h,\;\; j = 1, \ldots, N \right\},$$

where $h = (x_r - x_l)/(N-1)$; the number of points is set to $N = 1600$, which corresponds to approximately 10 points per wavelength. The 10-sub-domain grid is constructed from the single-domain grid by dividing the N points into ten

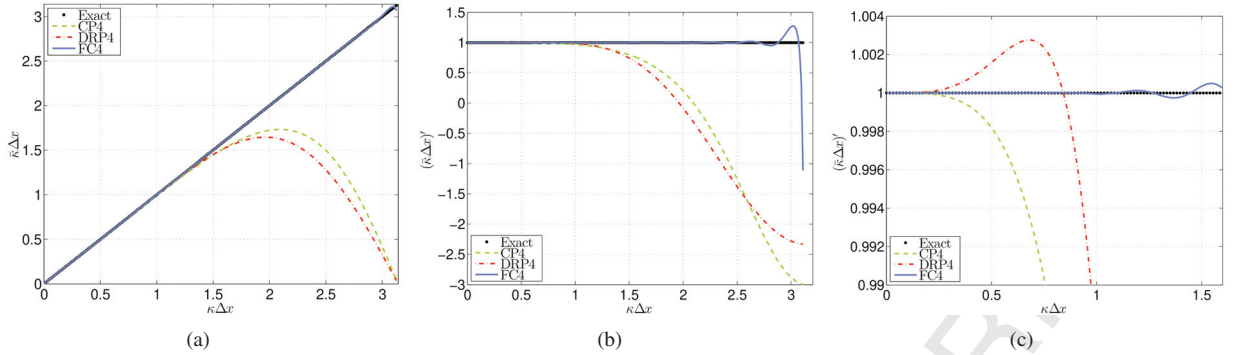


Fig. 2. Numerical wavenumber and numerical group speed of fourth-order approximations of the first derivative using Fourier continuation (FC4), a dispersion-relation-preserving finite difference scheme (DRP4), and compact finite differences (CP4). *a*) **Scaled numerical wavenumber $\bar{\kappa}h$ vs. scaled wavenumber κh.** *b*) **Scaled numerical group speed $(\bar{\kappa}h)'$ vs. scaled wavenumber κh.** *c*) **Close-up view of the numerical group speed $(\bar{\kappa}h)'$.**

overlapping sub-intervals in such a way that adjacent sub-domains share six points. Interval connectivity is enforced after each time step using the data exchange methodology described in Section 7.

Spatial derivatives are approximated using a fourth-order Fourier continuation (FC4) (see Remark 1 for details). For comparison, we also compute derivatives using a fourth-order optimized dispersion-relation-preserving (DRP4) finite difference scheme [28] (optimized over the scaled wavenumber range $0 \leq \kappa h \leq 1.1$) and a fourth-order compact (CP4) finite difference scheme [21] (using the notation of [21], the particular tridiagonal CP4 scheme that we use has parameters $\alpha = 1/4$, $\beta = 0, a = 2(\alpha + 2)/3$, $b = (4\alpha - 1)/3$, $c = 0$).

We perform a Fourier analysis to determine the resolution properties of each derivative approximation. Following [12], given a complex exponential $f(x) = e^{i\kappa x}$, we define the *effective* or *numerical wavenumber* $\bar{\kappa}$ by

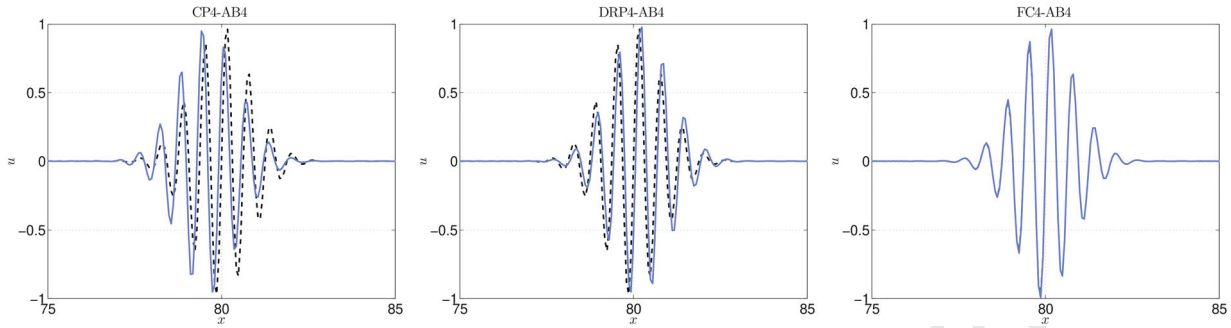
$$\bar{\kappa} := -i\frac{\langle Df, f\rangle}{\langle f, f\rangle}, \quad \text{where} \quad \langle f, g\rangle := \sum_{j=1}^{N} f(x_j)g^*(x_j) \tag{21}$$

and g^* denotes the complex conjugate of g. When the operator $D \equiv d/dx$, the numerical wavenumber $\bar{\kappa} = \kappa$. If $D = \tilde{D}$, an approximation to the spatial derivative operator, then $\bar{\kappa}$ is a function of κ. In the case where an equally-spaced grid is used, it suffices to restrict the *scaled wavenumber* κh so that $|\kappa h| \leq \pi$. In this case we also consider the scaled numerical wavenumber $\bar{\kappa}h$, a 2π-periodic function of κh, instead of just $\bar{\kappa}$. (An in-depth discussion of numerical wavenumbers of finite differences can be found in [40].)

The numerical wavenumbers of CP4, DRP4, and FC4 are shown in Figure 2(a). Note that although all three methods approximate the first derivative to fourth-order accuracy, there are important differences in their wave resolution properties. To 1% error, CP4 can resolve waves up to $\kappa h \approx 1.09$ with about six points per wavelength (PPW). DRP4 fares better at $\kappa h \approx 1.24$, requiring approximately five PPW to achieve the same accuracy. On the other hand, FC4 can resolve waves with $\kappa h \approx 2.12$ also at 1% accuracy but using only three points per wavelength.

Although numerical wavenumbers provide us with a measure of the accuracy of derivative approximations of a single complex exponential, it is well-known (see, e.g., [41]) that the wave propagation properties of a numerical method are more adequately described by the numerical group velocity (or group speed in the one-dimensional case). An expression for the group speed in terms of the wavenumber can be obtained from the dispersion relation. For this equation, if we substitute a solution of the form $v = e^{i(\kappa x - \omega t)}$ into (19), we find that the analytical dispersion relation is given by $\omega = \kappa$, where ω is the angular frequency; the analytical group speed is then $\frac{d\omega}{d\kappa} \equiv 1$.

Next, we derive an approximate numerical dispersion relation–under the assumption that the temporal error is relatively small compared with the spatial error–which suffices for our purposes. In all cases, the fully discrete solver is obtained from a combination of one of the three derivative approximations (CP4, DRP4, or FC4) and an explicit fourth-order Adams-Bashforth method (AB4). We use the relatively small value $\Delta t \approx 6.3 \cdot 10^{-4}$, for which we have $\Delta t/h = 0.01$, so that the main error contribution results from the spatial discretization. To fourth-order temporal



(a) (b) (c)

Fig. 3. Numerical solution of the advection equation (19) **over** $[0, 100]$ **at time** $T = 80$**. For reference, the exact solution (plotted using a dashed line) is also shown at the final time. The domain** $[0, 100]$ **is represented by a** 10 **sub-domain grid with a six-point neighbor overlap. In all cases the numerical solution is evolved to the final time** T **using fourth-order Adams-Bashforth (AB4). Derivatives are computed to fourth-order accuracy using a compact finite difference scheme (CP4), a dispersion-relation-preserving finite difference scheme (DRP4), or Fourier Continuation (FC4). The relative errors at time** T **are** *a*) $7.64 \cdot 10^{-1}$ **with CP4-AB4,** *b*) $5.81 \cdot 10^{-1}$ **with DRP4-AB4 and** *c*) $2.65 \cdot 10^{-2}$ **with FC4-AB4.**

accuracy, we obtain an approximate dispersion relation from

$$\frac{d}{dt}v^n \approx -\tilde{D}v^n \quad \Rightarrow \quad -i\omega v^n \approx -\tilde{D}v^n \quad \Rightarrow \quad \omega \approx -i\frac{\langle \tilde{D}v^n, v^n\rangle}{\langle v^n, v^n\rangle} = \bar{\kappa},$$

where $v^n := v(\cdot, n\Delta t)$ and the integer n denotes the time step. The *numerical group speed* is then approximately equal to $\frac{d\bar{\kappa}}{d\kappa}$, which we can also write in the following scaled form

$$(\bar{\kappa}h)' = \frac{d(\bar{\kappa}h)}{d(\kappa h)}. \tag{22}$$

Figure 2(b) shows that CP4 and DRP4 yield group speeds that are less than 1 for $\kappa h > 1.5$. FC4 reproduces the correct group speed to 1% accuracy for waves up to $\kappa h = 2.7$. With the wavenumber set to $\kappa = 10$ and the given resolution, we find that the wave packet's central wavenumber is $\kappa h \approx 0.63$. A magnified view of Figure 2(b), shown in Figure 2(c), predicts that waves with components at around $\kappa h = 0.63$ will propagate about 0.2% faster with DRP4, and approximately 1% slower with CP4, than the exact wave, which moves with speed one. In contrast, the group speed for FC4 predicts that the wave packet should propagate at the correct advection speed with an error of approximately $1.5 \cdot 10^{-6}$.

In the computational results that follow, we report relative errors ε_t at time t in the ℓ^∞-norm

$$\varepsilon_t := \frac{\|\mathbf{u} - \bar{\mathbf{u}}\|_\infty}{\|\mathbf{u}\|_\infty}, \quad \text{where} \quad \|\mathbf{u}\|_\infty = \max_{1\le j\le N}\{|u_j|\}, \tag{23}$$

and $\mathbf{u} = (u(x_1, t), \ldots, u(x_N, t))$ and $\bar{\mathbf{u}} = (\bar{u}_1, \ldots, \bar{u}_N)$ denote the exact and approximate solution to (20) at time t, respectively. For multi-domains, the error is defined to be the maximum relative error over all sub-domains.

Table 4. Relative errors for solution of the advection equation at time $T = 80$ **over 1- and 10-sub-domain grids. In all cases the numerical solution is evolved to the final time** T **using fourth-order Adams-Bashforth (AB4). Derivatives are computed to fourth-order accuracy using a compact finite difference scheme (CP4), a dispersion-relation-preserving finite difference scheme (DRP4), or Fourier Continuation (FC4).**

# sub-domains	1	10
CP4-AB4	7.63E-01	7.64E-01
DRP4-AB4	5.99E-01	5.81E-01
FC4-AB4	2.71E-02	2.65E-02

Solving to a final time $T = 80$ over the single-grid domain, the fully discrete solver CP4-AB4 propagates the center of the wave packet to $x = 79.3010$, for an approximate wave speed of 0.9913 or about 1% slower than the

exact advection speed. At $\kappa h = 0.63$, the CP4 group velocity predicts a speed of 0.9955, which is consistent with the numerical results. Over the 10-sub-domain grid, the results are similar: the wave travels to $x = 79.3007$ with nearly the same speed as in the single-domain case. From the DRP4 scheme's group velocity we expect a speed of 1.0026, so the numerical solution should overtake the analytical solution. The DRP4-AB4 computational results over the single- and multi-domain cases yield a speed error of $8 \cdot 10^{-4}$ and $7 \cdot 10^{-4}$, respectively; the wave having travelled to $x = 80.08$, ahead of the correct final wave location. The actual propagation speed error for FC4 over the 1- and 10-sub-domain cases, on the other hand, is $1.03 \cdot 10^{-5}$ and $7.25 \cdot 10^{-6}$, respectively. It is important to note the solution errors, displayed in Table 4, that accompany these propagation speed errors: as demonstrated in this table, the improved dispersion character of the FC approach translates into corresponding improvements in the solution error, both for the single and multi-domain algorithms, over the solution errors provided by the CP4 and DRP4 algorithms.

6. FC-based solver for variable coefficient ODE systems

The implicit time marching component of the implicit-explicit solver presented in this article requires solutions of one dimensional variable coefficient boundary-value ODE systems discretized by the FC spatial approximation. In this section, we develop an FC-based ODE system solver that is used in each sweep of the BDF-ADI implicit methods developed in Section 4.2.

Consider the ODE system

$$A\,q + B\,q_x + C\,q_{xx} = f \tag{24}$$

over the interval $x \in [0, 1]$, with general Robin boundary conditions

$$a\,q + b\,q_x = g, \quad x = 0 \tag{25a}$$

$$c\,q + d\,q_x = h, \quad x = 1 \tag{25b}$$

where q and f are m-dimensional vector-valued functions of x; A, B, and C are $m \times m$ matrix-valued functions of x; a, b, c, and d are $m \times m$ matrices; and g and h are m-vectors. The matrices C, a, b, c, and d could have rows equal to zero, and we do assume that each row of C is zero at $x = 0$ (respectively $x = 1$) if the corresponding rows of a and b (c and d) are both zero (i.e., a boundary condition is supplied for each row in equation (24) with second derivative terms).

Discretization of the interval into N equispaced points yields the vector- or matrix-valued grid functions $q_i = q(x_i)$, $A_i = A(x_i)$, etc. Let δ_x and δ_{xx} be the discrete FC first and second derivative operators, respectively. Then the system (24) is block-decomposed, with the i-th interior block ($i = 2, \ldots, N-1$) given by

$$A_i q_i + B_i \sum_{j=1}^{N} (\delta_x)_{ij} q_j + C_i \sum_{j=1}^{N} (\delta_{xx})_{ij} q_j = f_i. \tag{26}$$

The non-zero rows of the boundary conditions (25) are used to replace the rows of the discretized ODE for $i = 1, N$. In detail, let a superscript k denote the k-th row of a vector or matrix. We introduce the modified matrices and vectors $\hat{A}_i$, $\hat{B}_i$ and f_i ($i = 1, N$) given by

$$\hat{A}_1^k = \begin{cases} A_1^k & \text{if } a^k = b^k = 0 \\ a^k & \text{otherwise} \end{cases} \qquad \hat{A}_N^k = \begin{cases} A_N^k & \text{if } c^k = d^k = 0 \\ c^k & \text{otherwise} \end{cases}$$

$$\hat{B}_1^k = \begin{cases} B_1^k & \text{if } a^k = b^k = 0 \\ b^k & \text{otherwise} \end{cases} \qquad \hat{B}_N^k = \begin{cases} B_N^k & \text{if } c^k = d^k = 0 \\ d^k & \text{otherwise} \end{cases}$$

$$\hat{f}_1^k = \begin{cases} f_1^k & \text{if } a^k = b^k = 0 \\ g^k & \text{otherwise} \end{cases} \qquad \hat{f}_N^k = \begin{cases} f_N^k & \text{if } c^k = d^k = 0 \\ h^k & \text{otherwise} \end{cases}.$$

The first and N-th blocks of the discretized ODE system are then defined to be

$$\hat{A}_i q_i + \hat{B}_i \sum_{j=1}^{N} (\delta_x)_{ij} q_j = \hat{f}_i, \quad i = 1, N. \tag{27}$$

In general, the linear system defined by equations (26) and (27) is full and nonsymmetric, so we use a preconditioned GMRES iterative solver to invert the system [42]. A second order finite difference solver is used to left precondition the system, which greatly reduces the number of iterations needed for convergence [43]; cf. [29, Ch. 15.3]. The finite difference system is given by equations (26) and (27) with the discrete FC derivative operators δ_x and δ_{xx} replaced by corresponding centered difference schemes in the interior and a two-point one sided scheme at the boundary. The result is a block-tridiagonal system which we invert efficiently using a block-LU decomposition. The factorization need only be performed once per call of the GMRES solver.

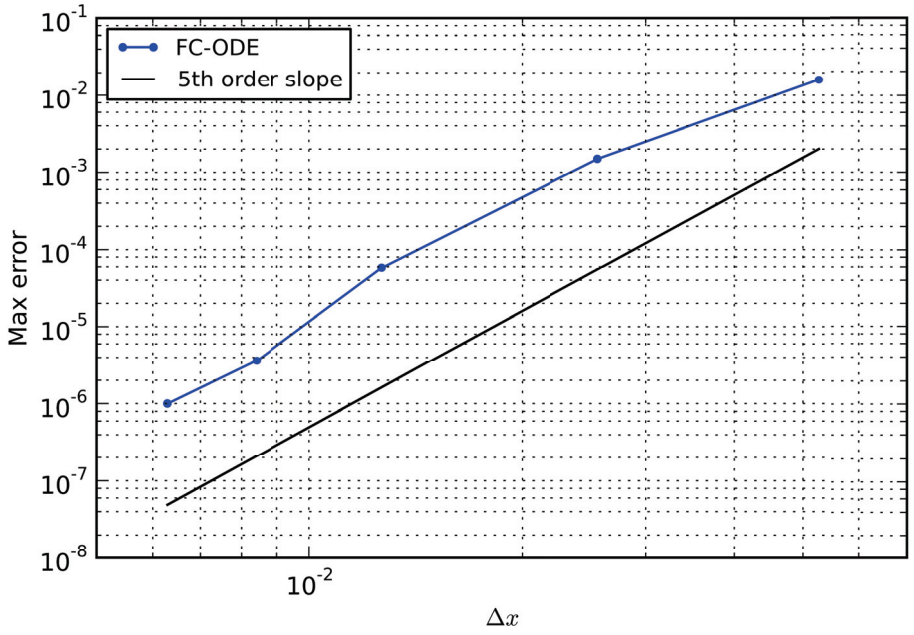


Fig. 4. One dimensional spatial convergence test of the variable coefficient FC-ODE solver for the system (28) **with exact solution** (29)**.**

Figure 4 demonstrates the convergence of the solver for the model ODE system

$$u + \Delta t\left(\widetilde{u}\,u_x + T_x + \frac{\widetilde{T}}{\widetilde{\rho}}\rho_x - \frac{\nu}{\widetilde{\rho}}u_{xx}\right) = f^1 \tag{28a}$$

$$T + \Delta t\left(\widetilde{u}\,T_x + \widetilde{T}\,u_x - \frac{\nu}{\widetilde{\rho}}T_{xx}\right) = f^2 \tag{28b}$$

$$\rho + \Delta t\,(\widetilde{u}\,\rho_x + \widetilde{\rho}\,u_x) = f^3 \tag{28c}$$

in the domain $x \in [0, 1]$, where $\Delta t = 0.01$ and $\nu = 0.01$ are constant. (Note that the form of these equations coincides with that of the system (16) that is solved in each sweep of the BDF-ADI scheme.) The functions $\widetilde{u}$, $\widetilde{T}$, and $\widetilde{\rho}$ are given by

$$\widetilde{u} = 1 - 6e^{\sin(5x)}, \quad \widetilde{T} = \log(3 + \sin(6x)), \quad \widetilde{\rho} = 1 + \frac{1}{2}\cos(4x^2 + 2), \tag{29}$$

and the right-hand side functions are chosen so that the exact solution is $u = \widetilde{u}$, $T = \widetilde{T}$, $\rho = \widetilde{\rho}$. Let $q = (u, T, \rho)^T$, then the boundary conditions are given by (25) with pseudorandom matrices

$$a = \begin{pmatrix} 0.71 & 0.39 & 0.19 \\ 0.51 & 0.71 & 0.06 \\ 0 & 0 & 0 \end{pmatrix} \quad b = \begin{pmatrix} 0.27 & 0.60 & 0.29 \\ 0.73 & 0.58 & 0.04 \\ 0 & 0 & 0 \end{pmatrix}$$

$$c = \begin{pmatrix} 0.87 & 0.57 & 0.20 \\ 0.19 & 0.41 & 0.51 \\ 0 & 0 & 0 \end{pmatrix} \quad d = \begin{pmatrix} 0.07 & 0.72 & 0.99 \\ 0.84 & 0.85 & 0.25 \\ 0 & 0 & 0 \end{pmatrix}$$

and where the right-hand sides are chosen to satisfy the exact solution. Notice that no boundary conditions are prescribed for ρ.

Remark 4. *The paper [43] presents an alternative variable coefficient scalar FC-ODE solver, which differs from the one presented here mainly in two respects: 1) It solves for the particular and homogeneous solutions separately, using the homogeneous solutions to correct the particular solution. This approach could be useful when solving many ODEs with the same coefficients and right-hand side but different boundary conditions. However, due to the nonlinearity of the Navier-Stokes equations, the ODEs we must solve have coefficients that change at each time step, and the approach in [43] provides no significant advantage. 2) It uses high-order asymptotic matching for the narrow boundary layers that occur in highly singularly perturbed ODEs, which were necessary to ensure stability of the PDE solver for very small Δt. These corrections were found to be unnecessary for the stability of the Navier-Stokes solver for all time discretizations we considered.*

7. Grid methodology

Our domain decomposition strategy relies on an overset (or composite) grid methodology to represent general domains for which a single smooth grid may be impossible or impractical to construct. Suppose that the domain of interest is a region $\Omega \subset \mathbb{R}^3$ with boundary $\partial\Omega$ and assume that there is a set of N^ν smooth transformations

$$(x^\nu(\xi,\eta,\zeta), y^\nu(\xi,\eta,\zeta), z^\nu(\xi,\eta,\zeta)) : [\ell_1,\ell_2]^3 \to \mathbb{R}^3, \qquad \nu = 1,\dots,N^\nu, \tag{30}$$

whose images cover the region and its boundary using N^ν subdomains:

$$\Omega \subseteq \bigcup_{\nu=1}^{N^\nu} \Omega_\nu \qquad \text{and} \qquad \partial\Omega \subseteq \bigcup_{\nu=1}^{N^\nu} \partial\Omega_\nu.$$

If $C_\nu := [\ell_1,\ell_2]^3$, ℓ_1, $\ell_2 \in \mathbb{R}$, denotes a reference Cartesian domain with coordinates (ξ,η,ζ), then each component subdomain

$$\Omega_\nu = \{(x^\nu(\xi,\eta,\zeta), y^\nu(\xi,\eta,\zeta), z^\nu(\xi,\eta,\zeta)) : (\xi,\eta,\zeta) \in C_\nu\}$$

is simply the image of C_ν under its associated mapping (x^ν, y^ν, z^ν). After defining a uniformly-spaced grid on C_ν

$$\begin{aligned}
\xi_j &= (j-1)\Delta\xi, \qquad j = 1,\dots,N^\nu_\xi, \qquad \Delta\xi = \frac{1}{N^\nu_\xi - 1},\\
\eta_k &= (k-1)\Delta\eta, \qquad k = 1,\dots,N^\nu_\eta, \qquad \Delta\eta = \frac{1}{N^\nu_\eta - 1},\\
\zeta_\ell &= (\ell-1)\Delta\zeta, \qquad \ell = 1,\dots,N^\nu_\zeta, \qquad \Delta\zeta = \frac{1}{N^\nu_\zeta - 1},
\end{aligned}$$

where N^ν_ξ, N^ν_η, and N^ν_ζ denote the number of nodes along each dimension, we apply the mapping (30) to obtain the "physical coordinates" of the νth grid

$$\mathcal{G}_\nu = \left\{\left(x^\nu_{jk\ell}, y^\nu_{jk\ell}, z^\nu_{jk\ell}\right) : x^\nu_{jk\ell} = x^\nu(\xi_j,\eta_k,\zeta_\ell),\ y^\nu_{jk\ell} = y^\nu(\xi_j,\eta_k,\zeta_\ell),\ z^\nu_{jk\ell} = z^\nu(\xi_j,\eta_k,\zeta_\ell)\right\}.$$

Grid connectivity is enforced through either piecewise polynomial interpolation in the general case, or using smooth, one-dimensional, convex combinations along grid lines when neighboring nodes overlap perfectly. Each overset grid shares a layer of points with neighboring sub-domains and has one of two types of overlapping boundaries: 1) the overlap region is imperfect, in that one or more points (generally all) do not correspond to any grid point in any neighboring grid, and 2) all points in the boundary region are also grid points in a neighboring patch, resulting in a perfect overlap. Boundaries of the first type are referred to as *interpolation boundaries*. They consist of sets of *interpolation points*, and they enable information transfer from one grid to another by means of suitably high-order (explicit) interpolation methods. The other kind of boundary, which we call an *exchange boundary*, does not require interpolation and data is simply exchanged among neighbors that share perfectly overlapping regions.

7.1. Interpolation boundaries

To describe our interpolation strategy, suppose that $p = (x^*, y^*, z^*)$ is an interpolation point in some domain $\Omega_{\nu'}$. By construction, p is contained in at least two patches. Suppose Ω_ν is one such patch and let $q = (\xi^*, \eta^*, \zeta^*)$ be the point in the computational domain of Ω_ν corresponding to p—that is, $x(\xi^*, \eta^*, \zeta^*) = x^*$ and similarly for the other coordinates. For Ω_ν to be an acceptable donor patch, an $m \times m \times m$ stencil of points (m is the stencil width, resulting in m-th order interpolation) must be found such that the stencil includes q in its interior in such a way that no points in the interpolation stencil used are themselves receivers of interpolation data from other patches. (Exchange points are not excluded from being part of a donor stencil to an interpolation point.) Of the stencils satisfying these requirements, the one with q closest to its center is chosen. A function is then interpolated in the computational domain using a tensor product Lagrange interpolation formula

$$u^{\nu'}_{j'k'\ell'} = \sum_{j=j_0}^{j_1} \sum_{k=k_0}^{k_1} \sum_{\ell=\ell_0}^{\ell_1} L_j(\xi_{j'}) L_k(\eta_{k'}) L_\ell(\zeta_{\ell'}) u^{\nu}_{jk\ell}. \tag{31}$$

To ensure the desired overall spatio-temporal accuracy order is obtained, the interpolation stencil is selected as described in [5]: the interpolation order of accuracy used is one more than that of the underlying spatial approximation. It is important to note that the grid overlap is kept constant as grids are refined (otherwise the interpolation order must increase accordingly as discussed in [5]). Extensive numerical experiments using complex, but fixed, overset grids indicate that this selection of interpolation stencils does not adversely affect either the high-order accuracy or the stability of the hybrid explicit-implicit overset solver in the present FC context. For dynamic overset grids, interpolation schemes that explicitly enforce global mass conservation can be advantageous [44].

7.2. Exchange boundaries

Exchange points can be thought of as interpolation points where the point p is itself one of the points in the donor stencil. For simplicity, in the remainder of this article "interpolation" boundaries (points) will refer to both interpolation and exchange boundaries (points). Differences in implementation will be noted where necessary. In all our examples, a stencil width of $m = 7$ is employed, and each interpolation boundary consists of a two-points-deep, so that the overlap at an exchange boundary is four points wide. Figure 5 illustrates the exchange data process in a one-dimensional case.

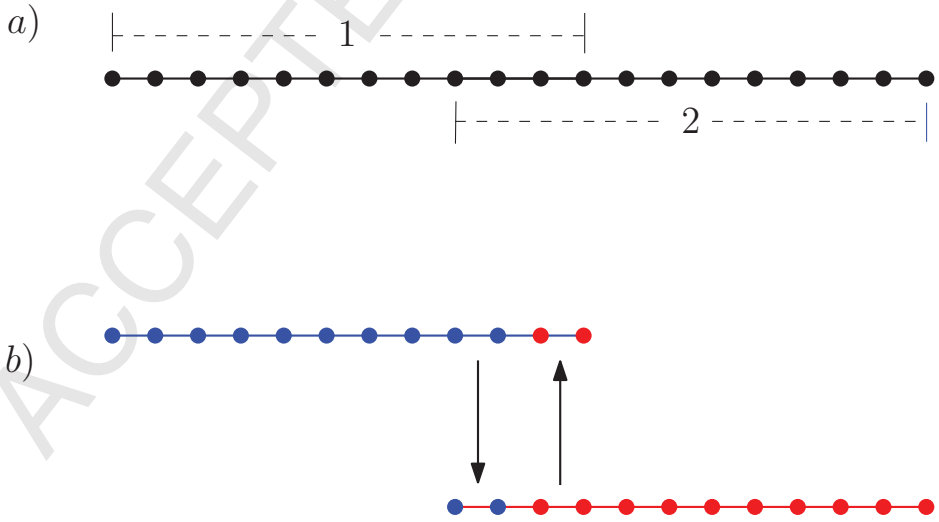


Fig. 5. One dimensional illustration of exchange boundaries. *a*) **Domains 1 and 2 overlap perfectly in a region four points wide.** *b*) **At the data-passing step of the algorithm, the solution values of the last two points in each domain are substituted by the corresponding values in the neighboring subpatch.**

8. Multi-domain implicit-explicit subiteration strategy

The proposed single-stage multi-domain implicit-explicit solver uses the same time-step value Δt for all patches. (Of course, this requires Δt to satisfy the strictest stability and accuracy requirements over all the component meshes. See also Section 4.1 for comments on possible use of multi-stage time-evolution.) Using this time step the algorithm proceeds as follows. At first the solution is evolved from time t^n to time $t^{n+1} = t^n + \Delta t$ over all explicit patches using the available physical boundary conditions as well as, when needed, values at time t^n on the patch boundary obtained from other explicit and implicit patches. The evolution from t^n to t^{n+1} over an implicit patch, in turn, requires boundary conditions at time t^{n+1} throughout the patch boundary—which generally includes boundary portions contained within the physical boundary, as well as portions that are contained within other explicit and implicit patches. The first two of these are readily available, either by evaluation of the given boundary conditions at time t^{n+1}, or by interpolation from t^{n+1} values obtained from the aforementioned explicit evolution. (In fact, a slight modification of this strategy is actually implemented in our solvers to improve parallel efficiency; see Remark 5.) The evaluation of last set of boundary conditions, finally, is obtained by means of (a small number of) subiterations which, starting from boundary values at time t^n to evolve the implicit patch, and subsequently exchanging information among all implicit patches and re-evolving to time t^{n+1} over the implicit patches, eventually produce the correct boundary values at time t^{n+1}. (In practice, as few as two subiterations suffice to produce the correct patch boundary condition starting from the solution at time t^n.)

In summary, the multi-domain implicit-explicit subiteration algorithm proceeds as follows:

1. All explicit patches are marched forward in time from time t^n to time t^{n+1}.
2. Interpolation data from explicit patches is sent to neighboring patches.
3. All implicit patches are iteratively marched forward in time with exchange of boundary values at the end of each subiteration, to eventually obtain the necessary boundary values at time t^{n+1} (See remark 5).
4. The subiteration process continues until a prescribed number of subiterations have been completed, which completes the evaluation of the solution at time t^{n+1}.

Remark 5. *Steps 1 and 3 could be performed in parallel provided approximate (e.g., extrapolated) boundary conditions are used. This approach cannot be carried for all subiterations without leading to instability but under certain circumstances this strategy is effective. For example, a processor assigned to an implicit zone can proceed with the first subiteration in parallel with the explicit solves. In this case, an initial approximation for the boundary conditions is obtained by means of the temporal extrapolation formula* (13)*. In practice this approach has provided an effective parallelization methodology.*

9. Parallelization

The overset mesh framework together with the parallel subdomain subiteration strategy naturally lead to a parallel implementation in a distributed computing environment. An efficient implementation should divide the workload among all processors as equally as possible. Details in this regard for explicit FC solvers is well documented in [12, 13]. Additional details, concerning parallel implementation and load-balancing in the implicit-explicit context, are presented in what follows. To this end, this section provides heuristics for decomposing a domain to maximize efficiency, as well as an investigation of the computational cost of the implicit multi-domain approach.

9.1. Implicit multi-domain load balancing

The present load-balancing algorithm is based on those presented in [12, 13]. Given a set of implicit patches $\{\Omega_j^I\}$, $j = 1, \ldots, M_I$ and a number of target patches p_{total}, the target number of sub-patches assigned to an implicit patch Ω_j^I is

$$p_j^I = \frac{N_\xi^j N_\eta^j N_\zeta^j}{\sum_{l=1}^{M_I} N_\xi^l N_\eta^l N_\zeta^l} p_{\text{total}}$$

(rounded to the nearest integer), where N_ξ^j, N_η^j, N_ζ^j are the numbers of discretization points in each dimension. The partitioning of a patch into subpatches is performed in such a way that approximately the same number of discretization points is used along the various spatial dimensions. If p is the number of parallelization sub-patches assigned to a patch, then the number of partitions in the ξ direction is given by

$$s_\xi = N_\xi \left(\frac{p}{N_\xi N_\eta N_\zeta} \right)^{\frac{1}{3}},$$

rounded to the nearest integer greater than or equal to one. Similar formulas hold for the number of partitions in the η and ζ directions.

9.2. Implicit multi-domain performance

In this section, we present the results of various computational tests of the parallel implicit multi-domain solver in a distributed computing environment. All results were obtained on a Poweredge cluster with Infiniband networking, consisting of 32 compute nodes, each of which has two eight-core Intel Xeon E5-2665 processors (for up to 32 threads per node with Intel Hyper-threading) and 64 GB of memory.

In [12, 13], the authors quantified the cost of the explicit FC solvers using the number of seconds S required per processor to advance one million unknowns forward one time step, which, for our Navier-Stokes solver, is given by the formula

$$S = \frac{(\#\text{ of processors}) \times (\text{total compute time per step}) \times 10^6}{(d+2) \times (\#\text{ of discretization points})}, \tag{32}$$

where the factor of $d+2$ in the denominator is the size of the vector of unknowns Q in $d = 2$ and 3 spatial dimensions. The present explicit FC-based Navier-Stokes solver achieves approximately the same efficiency as the solvers in those contributions—namely,

$$S \approx 1.6 \text{ sec.} \quad \text{(for explicit solvers)}. \tag{33}$$

To test the efficiency of the parallel implicit algorithm, we use a smaller version of the mesh described in Section 10.3, which is used for three dimensional tests of flow past a sphere. In this example, the limits of the domain are $[-3, 3]$ in each dimension, with the extent of the quasi-spherical curvilinear patches the same as in Section 10.3. The mesh spacing in the Cartesian patches is a constant value h, while the radial spacing in the curvilinear patches is $h/10$ near the surface of the sphere and h at the outer edge of the patch. The initial conditions and source terms are the same as in the tests of flow past a sphere.

$h_{\max}$	0.06	0.048	0.04		
# grid points	2,045,990	3,817,125	6,372,454		
# sub-domains	104	194	104	194	294
S_1	36.8	33.4	33.9	30.7	32.6
S_2	54.3	58.9	53.4	51.6	55.3
S_3	86.6	78.8	74.4	74.9	77.0

Table 5. Number of seconds S_j, $j = 1, 2, 3$, per time-step, per processor, and per million unknowns required by the parallel implicit algorithm with various numbers of discretization points and sub-domains, and using j subiterations.

Table 5 reports the value of S for the BDF2-ADI solver on the composite mesh for various values of h, sub-domain partitionings, and number of sub-iterations. In these tests, the Mach and Reynolds numbers were taken to equal 0.8 and 10^3, respectively, and the GMRES residual tolerance was set to 10^{-6}. The solver was run for a total of 20 time steps with $\Delta t = 10^{-3}$. The results show that S is approximately constant for each fixed number of sub-iterations, which corresponds to approximately linear parallel scaling of the implicit algorithm. Extensive experimentation has shown that as few as two iterations suffice to obtain convergence up to the error level otherwise inherent in the discretization, and that no accuracy improvements results from use of additional subiterations.

We can also use the data in Table 5 to compare the computational costs of the explicit and implicit algorithms. For one sub-iteration, the implicit solver is about twenty times more expensive than the explicit solver—which is reasonable if, for example, each ADI sweep in each of the three spatial dimensions takes about six or seven GMRES iterations to converge. Two sub-iterations of the implicit solver (the number we use in our numerical examples)

is about thirty-five times more expensive than the explicit solver. This suggests that the implicit solver is most advantageous when the time step Δt required by the explicit solver for stability is about 35 times smaller than that required by the implicit solver.

Remark 6. *In the context of the implicit solver presented in this article, many factors can affect the convergence rate of GMRES—such as the size of the time step Δt, the physical parameters (particularly Reynolds and Mach numbers), how well the solution is spatially resolved, the accuracy order of the temporal discretization, boundary conditions, etc.—but we have found the most influential factors to be the time step Δt and the Reynolds number. As $\Delta t \to 0$, the operator to be inverted is asymptotic to the identity, leading to faster convergence. On the other hand, large Reynolds number makes the operator more singularly perturbed (i.e., the coefficients of the second derivatives become smaller), and the spectrum of the centered finite difference preconditioner does not approximate the spectrum of the actual operator well (see [29, Ch. 15.3]). Nevertheless, we have observed that the performance results given in Table 5 are typical.*

Remark 7. *A rigorous analysis of semi-discrete BDF methods (continuous in space, discrete in time) applied to the advection diffusion equation*

$$u_t + \alpha u_x = \beta u_{xx}$$

on the real line divided into two overlapping subdomains is included in [39, Sec. 3.4.1]. There it is shown that the approximate number of subiterations needed to reduce the error to a tolerance on the order of truncation error, $O((\Delta t)^{s+1})$, is $\approx \frac{-\ln \Delta t}{\bar{\lambda}\delta}$, where δ is the the width of the overlap between the two subdomains and $\bar{\lambda} = \sqrt{\frac{\alpha^2}{4\beta^2} + \frac{1}{\beta b \Delta t}}$. In practice we have generally not observed improvement in solutions of the Navier-Stokes equations beyond two subiterations.

10. Numerical results

This section showcases the full multi-domain implicit-explicit algorithm with examples in two and three spatial dimensions. All of the results presented in this section were obtained on the computer cluster described in Section 9.2. The main purpose of the simulations is to showcase the accuracy and stability of the proposed methodologies. With regards to accuracy, Figures 7 and 11 in this section present high-order convergence slopes that are closely in line with the corresponding theoretical estimates. A theoretical discussion of stability in the multidomain context has not been attempted—although the stability of the individual BDF-ADI-based and AB-based Navier-Stokes solvers has been studied and demonstrated extensively in [12, 11, 17]. Additionally, the extensive numerical experiments presented in this section, which have required evolution of the numerical solution up to many hundreds of thousands of time-steps, for both manufactured-solution and fully physical solutions, do indicate the numerical stability of the proposed overall multi-domain implicit-explicit methodology. Once again we emphasize that these results represent the first numerical demonstrations of high-order time-accuracy for any multi-domain solver of the compressible Navier-Stokes equations. Unless otherwise indicated, all simulations use the parameter values Pr = 0.71 and $\gamma = 1.4$, and the (non-dimensional) viscosity and thermal conductivity are given by Sutherland's law (equation (2)) with $S_\kappa = S_\mu = 0.3$.

10.1. Unsteady flow past a cylinder

This section presents numerical results for the problem of flow past a cylinder in two spatial dimensions with Re = 200. At higher values of the Reynolds number three-dimensional effects become important thus reducing the relevance of two-dimensional simulations. The two-dimensional cross-section is the region inside the rectangle $[-8, 24] \times [-8, 8]$ and outside a circle of radius $r_c = 0.5$ centered at the origin. This region is divided into 13 overlapping patches: four curvilinear annular-like sections surrounding the cylinder and nine rectangular patches away from the cylinder. Figure 6 shows a portion of the domain including one of the discretizations used. Given upper and lower bounds $h_{\max}$ and $h_{\min}$ on the acceptable spatial grid size, the meshes are generated in such a way that the radial spacing is guaranteed to be $\geq h_{\min}$ near the surface of the cylinder and $\leq h_{\max}$ at the outer edge of the annular sections. The mesh spacing in rectangular Cartesian domains is approximately $h_{\max}$. To ensure this, the number of discretization points in the x direction of a Cartesian domain, for example, is taken to equal $1 + \frac{L}{h_{\max}}$ rounded to the

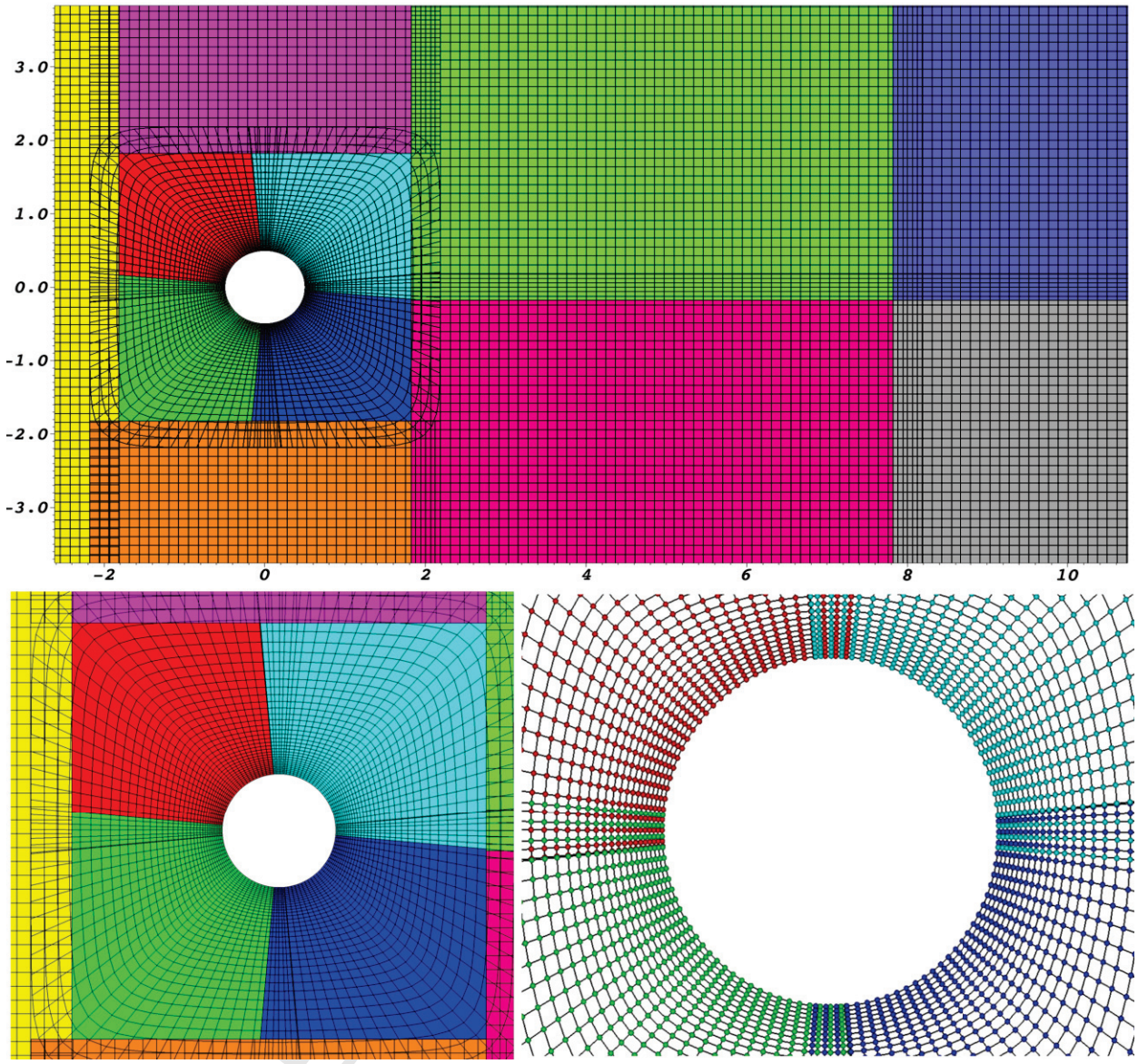


Fig. 6. Three close-up views of the mesh used in the numerical experiments of flow past a cylinder. (For clarity, only every other grid point is plotted.) The top figure shows overlapping annular regions where implicit time-stepping is used; the solution in the remaining background Cartesian domains is evolved explicitly. The bottom left figure shows the clustering of points near the cylinder surface to spatially resolve the boundary layer. The bottom right figure shows the subdomain overlaps and associated interpolation layers.

nearest integer, where L is the side of the rectangle in the x direction. In this example the four curvilinear patches use the BDF-ADI algorithm to march the solution forward in time, while the rectangular patches use explicit Adams-Bashforth time-marching.

For this example we have assumed no-slip and isothermal ($T = 1$) boundary conditions at the boundary of the cylinder. At the outer edges of the domain, an absorbing layer is used. In detail, let $\partial^e\Omega$ denote the outer boundary of the domain Ω (i.e., not including the boundary located at the surface of the cylinder). For an absorbing layer of width w_a, the matrix M^0 in the quasilinear-like formulation (8) is given by $M^0 = \sigma(x, y)I$, where I is the identity operator and σ is the function

$$\sigma(x,y) = \begin{cases} A_a \left(1 - \frac{d((x,y),\partial^e\Omega)}{w_a}\right)^{p_a} & d((x,y),\partial^e\Omega) < w_a \\ 0 & \text{otherwise,} \end{cases} \tag{34}$$

where $d((x,y),\partial^e\Omega)$ is the distance from the point (x,y) to the boundary $\partial^e\Omega$, and where, for definiteness, we have selected the constants $A_a = 5.0$ and $p_a = 4$. The source term σQ_{ref}, in turn, is added to the right-hand side of equation (8), where Q_{ref} is the vector of unknowns corresponding to the initial conditions (35).

For our purposes, the initial condition and right-hand side source terms for the equations satisfy three requirements: 1) They must start the simulation in such manner that the time derivative of all the fields at $t = 0$ is 0 (so that the initial condition can be used to initialize all the previous time steps of the multi-step time marching schemes, without reducing the temporal order of accuracy), 2) they reduce the amount of time it takes for the solution to settle into the periodic vortex shedding mode by breaking the symmetry of the flow early on and 3) they satisfy the no-slip, isothermal boundary condition at the surface of the cylinder and the free stream conditions in the absorbing layer for all times t. In view of these goals the flow is initialized with the radially symmetric fields

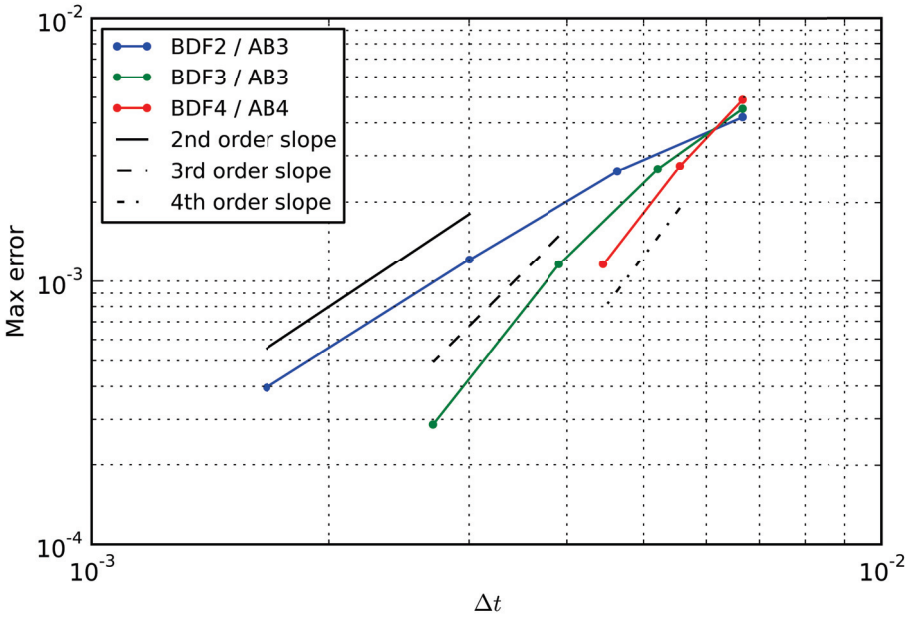


Fig. 7. Temporal convergence of the solver for flow past a cylinder at time $t = 1.0$, with $\text{Re} = 200$ and $\text{Ma} = 0.8$.

$$\rho = T = 1, \quad v = 0, \quad u = u_0\psi_s(r - r_c), \tag{35}$$

where u_0 is the freestream velocity and ψ_s is the smooth step function

$$\psi_s(x) = \begin{cases} 0 & x \le 0 \\ \left(1 + \exp\left(\frac{1}{x} - \frac{1}{1-x}\right)\right)^{-1} & 0 < x < 1 \\ 1 & x \ge 1. \end{cases} \tag{36}$$

This initial condition satisfies the no-slip and isothermal conditions at the surface of the cylinder as well as the uniform free-stream conditions in the absorbing layer. A source term is added to the right-hand side of the equations that enables the initial flow to transition smoothly from the initial condition given above on to the truly physical regime for which the right-hand-side resulting from use of a manufactured solution actually vanishes. The transition is effected by smoothly taking the right-hand side source term to zero over the time interval $0 \le t \le t_0$ by multiplying it by $\psi_s(1 - t/t_0)$ for some fixed time constant t_0, allowing the physics of the configuration to evolve the solution thereafter. This ensures the solution is smooth in time at $t = 0$. To break the symmetry of the flow, we include an additional source term on the right-hand side of the v equation:

$$\psi_w(t; 0, 0.2, 0.8, 1) f_G(x, y, t),$$

where ψ_w is the smooth windowing function

$$\psi_w(x; a, b, c, d) = \psi_s\left(\frac{x-a}{b-a}\right)\psi_s\left(\frac{d-x}{d-c}\right), \quad a < b \le c < d \tag{37}$$

and f_G is the Gaussian pulse

$$f_G(\mathbf{x}, t) = A_0 e^{\frac{|\mathbf{x}-\mathbf{x}_0|^2}{2\sigma_0^2}} \sin(\omega_0 t), \tag{38}$$

with parameters $\mathbf{x}_0 = (1.5, 1)$, $A_0 = -0.4$, $\sigma_0^2 = 0.25$, and $\omega_0 = 2\pi$.

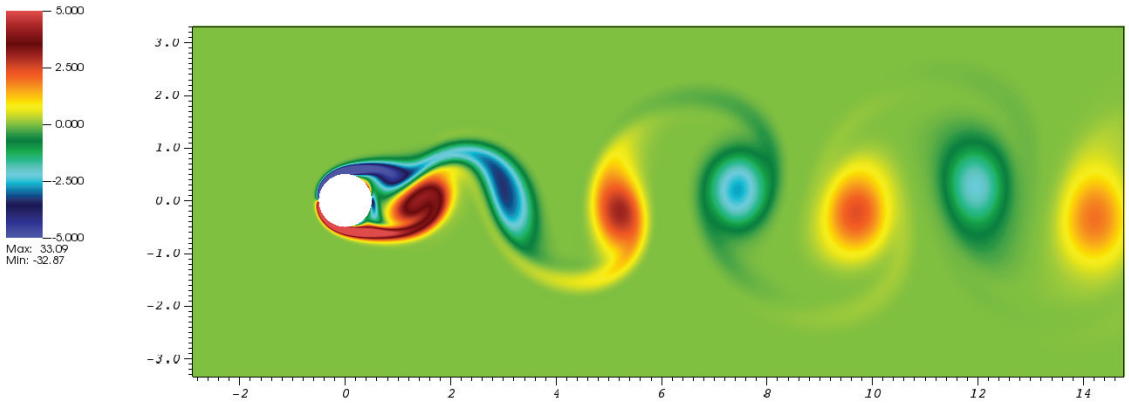


Fig. 8. Snapshot of the vorticity in a simulation of flow past a cylinder with Re = 200, Ma = 0.2 **at time** $t = 82.8$.

Having completed the set-up details for the evaluation of the flow past a cylinder we now present a variety of numerical results that illustrate the properties of the proposed methods. We first consider the convergence of the solver as h and Δt are simultaneously refined. For this test, we use Re = 200 and Ma = 0.8 and the time over which the manufactured initial condition is phased out is $t_0 = 0.5$. The value Ma=0.8 was used to facilitate resolution of the acoustic waves while utilizing sufficiently coarse time discretizations to easily allow for meaningful convergence studies. At this value of the Mach number a transonic shock develops for sufficiently long flow time, but our test run was stopped well before the viscous shock forms. The mesh size values $h = h_{\max} = 0.06, 0.05, 0.04, 0.03$, and 0.02 are used for the Cartesian domains and $h_{\min} = 0.1h$ is used at the cylinder surface. In view of accuracy considerations, the time step Δt is chosen to be of the form $\Delta t = Ch^p$ for some power p depending on the order of the method: $p = 2, 1.3, 1$ for $s = 2, 3, 4$ respectively (see Remark 8 below). The constant C is given by $C = 0.06^{-p}/150$, so that $\Delta t = 1/150$ when $h = 0.06$. Figure 7 plots the L^∞ error at time $t = 1.0$ versus Δt confirming the expected temporal order of accuracy of the solvers. The error is estimated by comparing against the solution obtained on the finest grid ($h = 0.02$) with $\Delta t = 1/800$ using fourth-order BDF-ADI and AB4 time marching.

Next, we show the results of a full simulation of unsteady flow past a cylinder with Re = 200 and Ma = 0.2. (We use a lower Mach number in this demonstration to more closely approximate incompressible flow.) For this test, the mesh we use has $h_{\max} = 0.06$ and $h_{\min} = 0.006$. The manufactured initial condition and v velocity source term are phased out at $t_0 = 5.0$. Third order time-marching is used in all patches—BDF3-ADI in the implicit patches and AB3 in the explicit patches. The time step is chosen to be $\Delta t = 3.33 \cdot 10^{-3}$ and the simulation is run for 30,000 time steps. Figure 8 shows the characteristic von Karman vortex street at time $t = 82.8$. The Strouhal number St (the non-dimensional frequency of vortex shedding) is found by tracking the vertical velocity at the point $(0, 1)$, and is estimated to be St = 0.202, which is consistent with the results reported in [12] and experimental references therein for the same Reynolds number and a slightly different Mach number. In Figure 9, snapshots of the streamlines near the cylinder show the time evolution of vortex shedding.

Remark 8. *The total error E in a simulation is composed of the errors associated with the spatial and temporal approximation, so that $E \sim h^r + (\Delta t)^s$. If only Δt or h are refined, then eventually the error will be dominated by the*

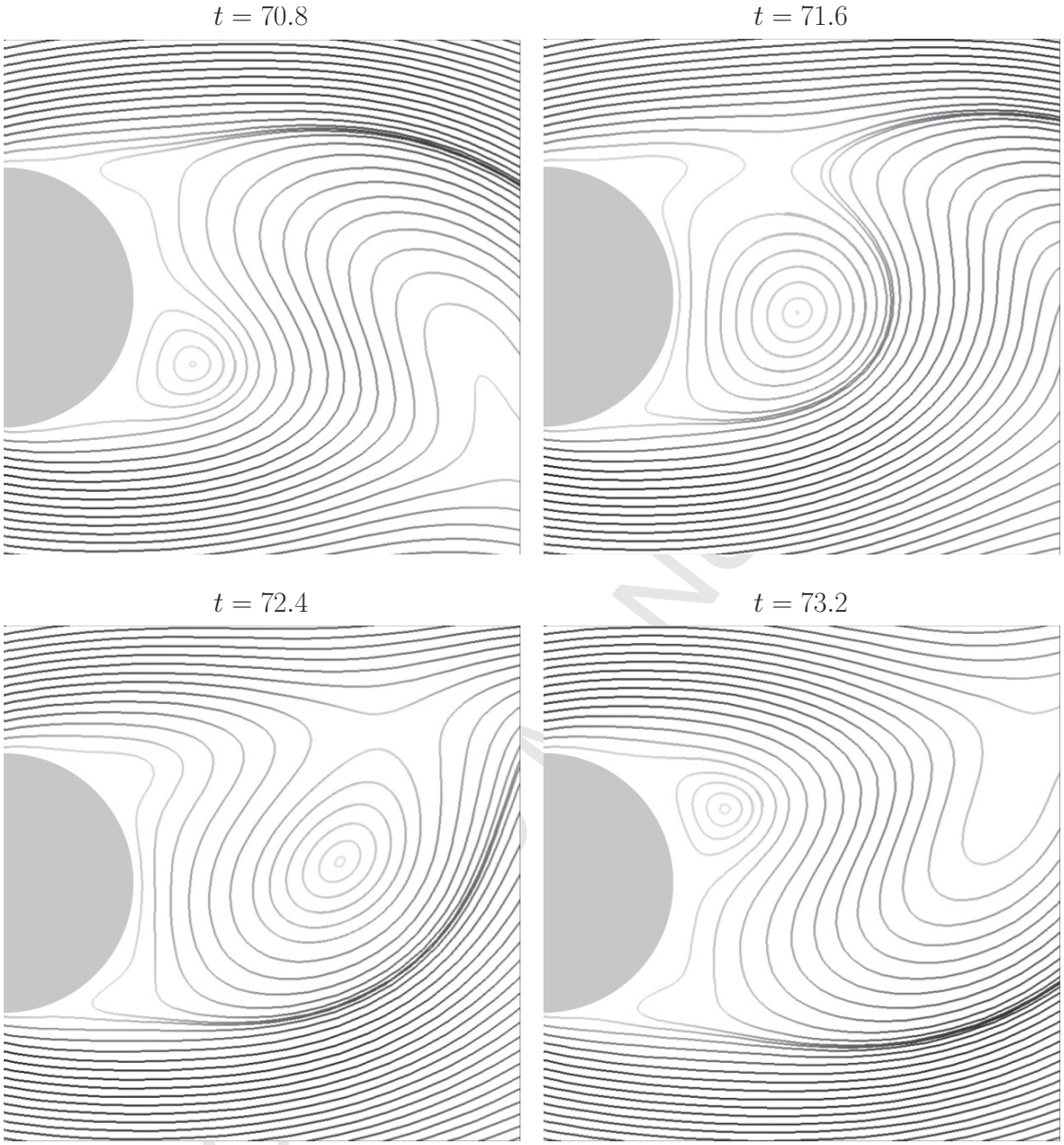


Fig. 9. Time evolution of streamlines in flow past a cylinder at Re = 200 **and** Ma = 0.2**. Darker shading of the streamline corresponds to a higher magnitude of the velocity at that point.**

other contribution. In order to see a convergence slope of $(\Delta t)^s$ *we must have* $E \sim (\Delta t)^s$ *for the total error, which implies a constraint on the spatial discretization of* $h \lesssim (\Delta t)^{s/r}$*, or equivalently* $\Delta t \gtrsim h^{r/s}$*. In other words, to see the spatio-temporal convergence slope with simultaneous refinement in space and time,* Δt *cannot be too small relative to* h*.*

10.2. *Unsteady flow past an airfoil*

Figure 10 presents results of a numerical simulation for flow past a NACA0012 airfoil in two spatial dimensions with Re = 500 and Ma = 0.5, obtained by means of the BDF2-AB2 implicit explicit solver. Note the successive levels of refinement used to resolve the viscous boundary layer, and the very fine meshes near the airfoil boundary—which are adequately treated by means of the implicit BDF2-ADI scheme.

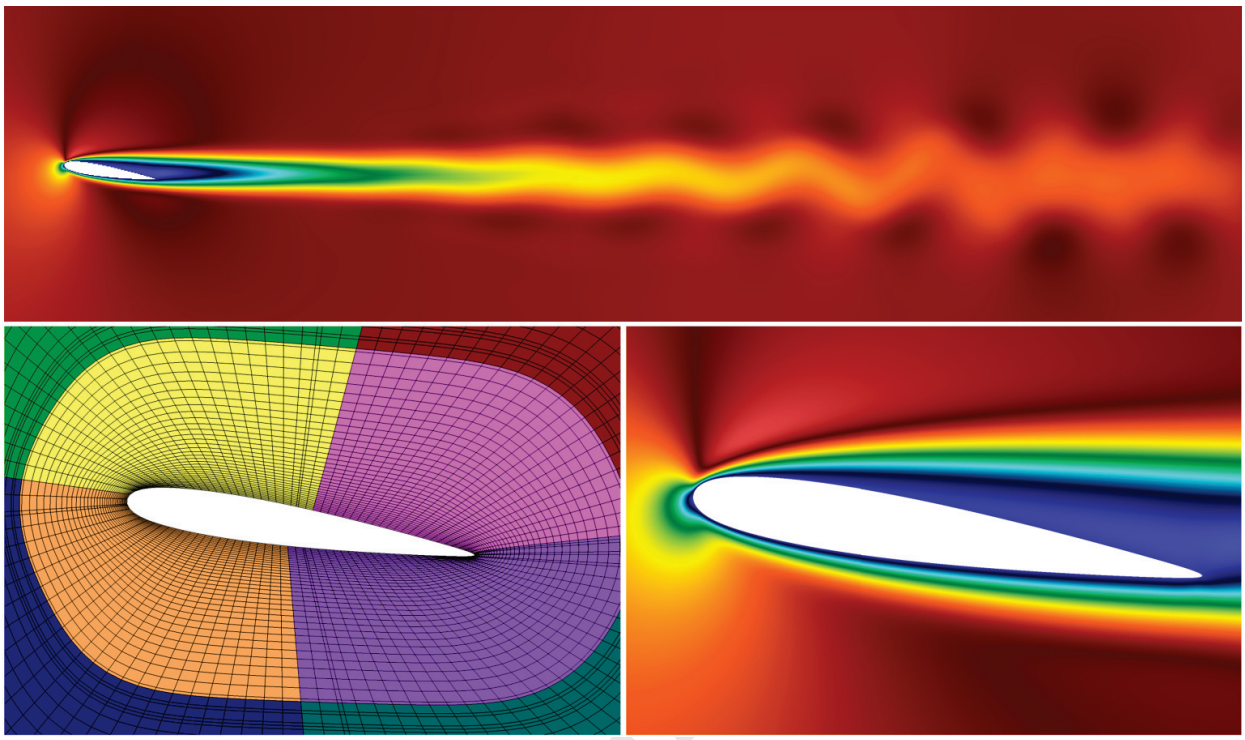


Fig. 10. Flow past a NACA0012 airfoil at a 3° angle of attack, with $\mathrm{Re} = 500$ and $\mathrm{Ma} = 0.5$. The top figure shows a snapshot of the x-velocity at time $t = 20$. The bottom left figure shows a close-up of the overset grids which are successively refined near the obstacle to resolve the boundary layer. (For clarity, only every fourth grid point is plotted.) The bottom right figure shows a close-up view of the x-velocity around the airfoil.

10.3. Unsteady flow past a sphere

Q_j	α_j	β_j	λ_j	$\phi_{j,t}$	$\phi_{j,x}$	$\phi_{j,y}$	$\phi_{j,z}$
u	0	1	7(37)	−1	0	0	0
v	0	1	7(37)	−2	0	0	0
w	0	1	7(37)	−3	0	0	0
ρ	1	0.2	7(37)	−4	4	7	14
T	1	0.2	7(37)	−5	5	6	15

Table 6. Parameters for the three-dimensional manufactured solution. The temporal frequencies λ_j not in parentheses are the ones used in convergence tests for methods of orders $s = 2, 3$, while those in parentheses are used for $s = 4, 5, 6$.

In this section, we present the results of three-dimensional flow past a sphere. The domain is the region inside the box $(x, y, z) \in [-4, 14] \times [-4, 4] \times [-4, 4]$ and outside the sphere of radius $r_s = 0.5$ centered at the origin. This region is divided into 12 larger overlapping patches: two curvilinear patches forming a modified "Yin-Yang" mesh [45] surrounding the sphere and ten Cartesian patches away from the sphere. The BDF-ADI method is used in the curvilinear patches and explicit AB time marching is used in all Cartesian patches. Given values h_s, h_c, h_f, and h_u, to be selected in what follows, meshes are generated in such a way that the radial spacing is h_s near the surface of the sphere, h_c at the edge of the Yin-Yang meshes and in the Cartesian patches directly downstream from the sphere, h_f in the far-field downstream Cartesian patches, and h_u in the Cartesian patches that are not downstream from the sphere.

We use the method of manufactured solutions to demonstrate the temporal convergence of the second and third order solvers at $\mathrm{Re} = 500$ and $\mathrm{Ma} = 0.8$. For this test, we use a smaller domain $[-4, 4]^3$ and set the spatial mesh-size

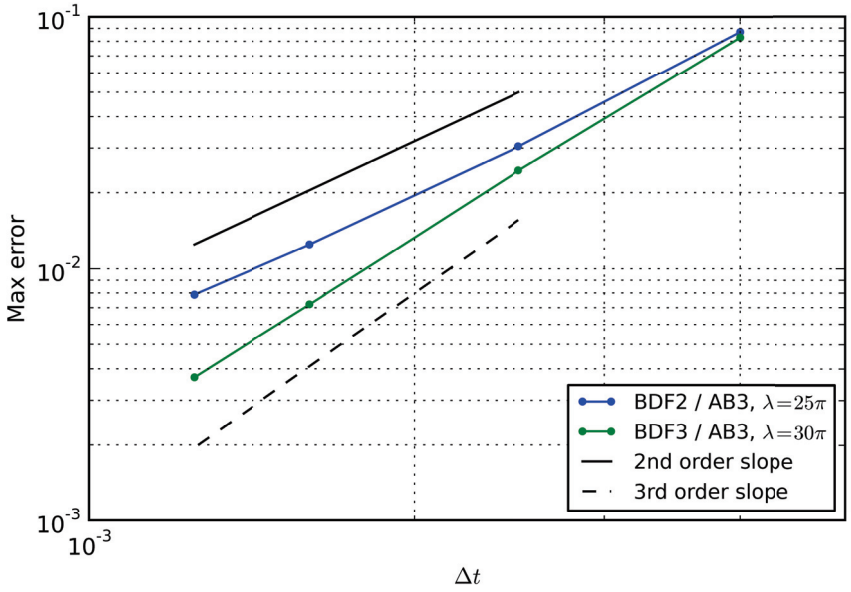


Fig. 11. Temporal convergence of the three-dimensional multi-domain solver using the method of manufactured solutions at time $t = 1.0$, with Re = 500 and Ma = 0.8.

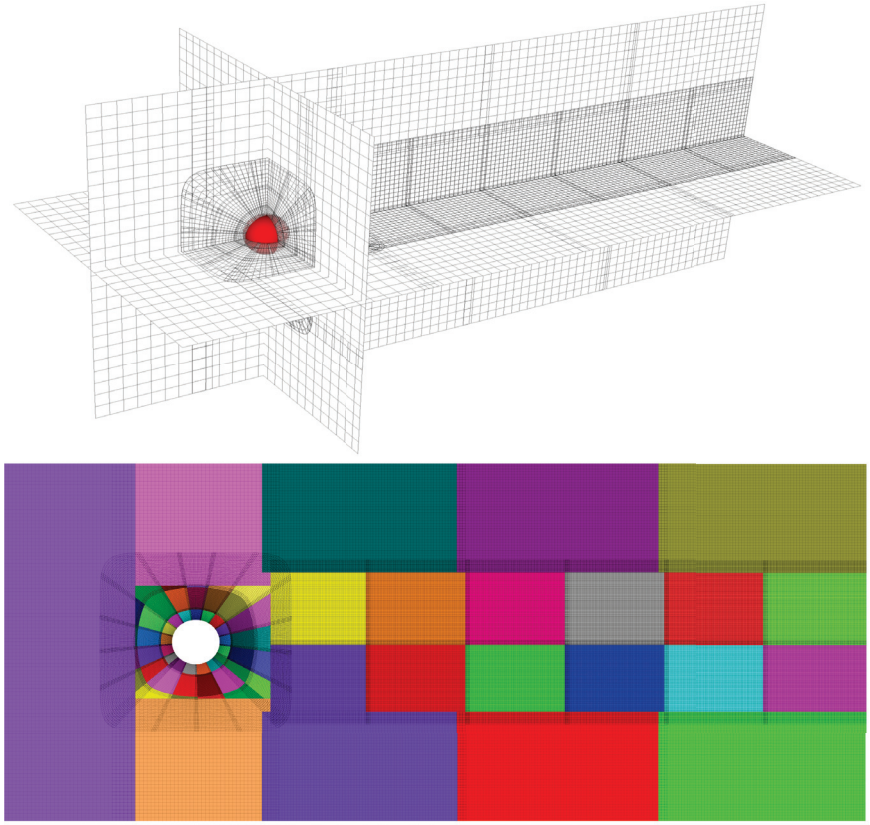


Fig. 12. Upper image: a three-slice of the mesh used for flow past a sphere. (For clarity, only every fourth grid point is plotted.) Lower image: a mesh cross-section showing the sub-patch decomposition with refinement in the wake region.

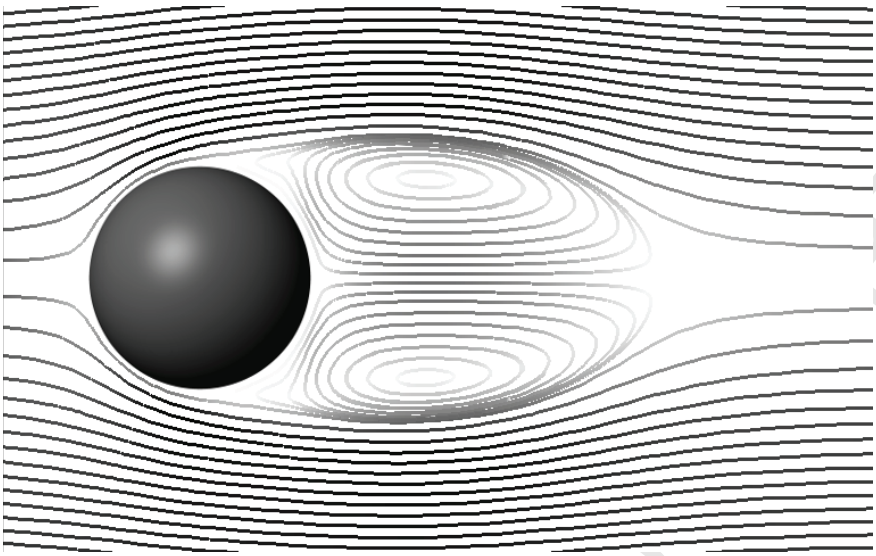
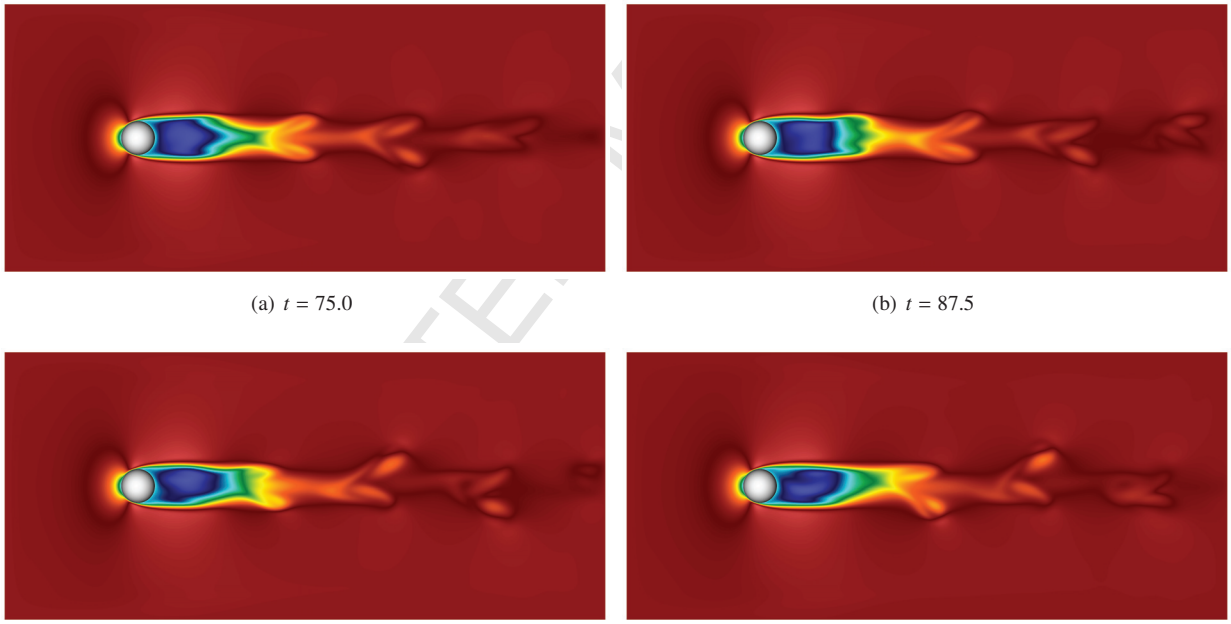


Fig. 13. Two-dimensional x-z slice of the streamlines in a simulation of flow past a sphere with Re = 500**,** Ma = 0.5 **at time** $t = 12$**. Darker shades in the streamlines indicate higher velocity magnitude.**



(a) $t = 75.0$ (b) $t = 87.5$

(c) $t = 100.0$ (d) $t = 112.5$

Fig. 14. Time evolution of the x-velocity for flow past a sphere at Re = 500 **and** Ma = 0.5 **illustrating the asymmetric development of vortical structures.**

parameters to be $h_s = 0.004$, $h_c = h_f = h_u = 0.04$. The manufactured solution is given by the equation

$$Q_j(x,t) = \alpha_j + \beta_j \sin\left(\lambda_j t + \phi_{j,t}\right) \prod_{i=1}^{3} \sin\left(2\pi x_i + \phi_{j,x_i}\right), \tag{39}$$

where Q_j is the jth component of the solution vector and where α_j, β_j, λ_j, ϕ_{j,x_i} are the constants shown in Table 6, except for λ_j which is set to $\lambda_j = 25\pi$ and $\lambda_j = 30\pi$ for the second and third order solvers respectively. Dirichlet

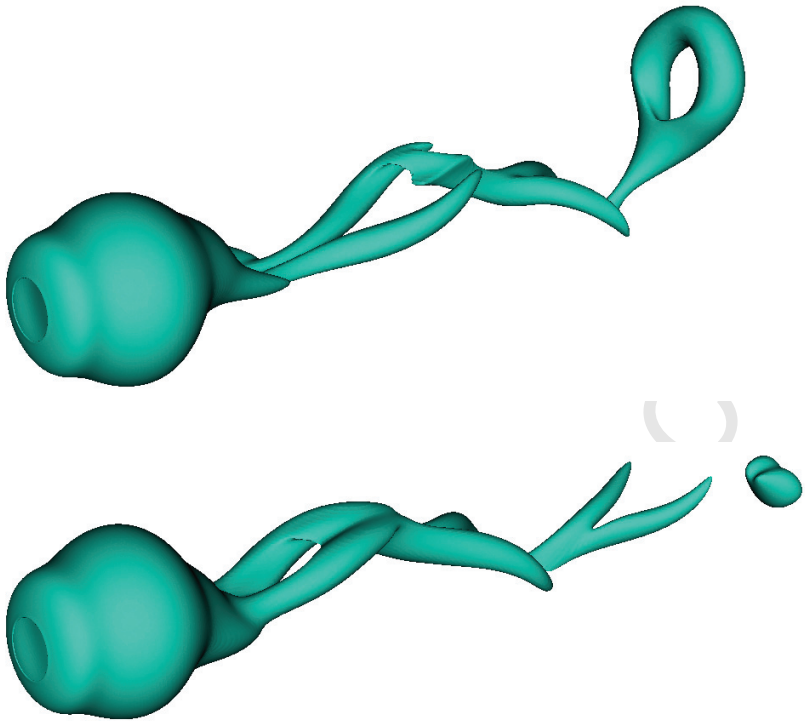


Fig. 15. Isosurfaces of the density ($\rho = 0.95$) in three-dimensional flow past a sphere at times $t = 64.5$ (top) and $t = 77.5$ (bottom), showing the appearance of hairpin vortices in the flow field.

boundary conditions are used for the velocity and temperature on the surface of the sphere and, as in the physically valid applications to the Navier-Stokes equations, no boundary condition is enforced for the density: this quantity is evolved according to the discretized PDE up to and including points on the boundary, and the boundary density data, which is available from the expressions for the manufactured solution, are not utilized in any way as part of the numerical solution process. Figure 11 shows nearly perfect second and third order convergence slopes.

Next, we provide a physical example of flow past a sphere at Re = 500 and Ma = 0.5. The mesh size parameters in this example are $h_s = 0.005$, $h_c = 0.3$, $h_f = 0.6$, and $h_u = 0.08$. The 12 larger patches are divided into 365 subpatches and one processor is assigned to each subpatch. The number of points in each subpatch is such that the costs of one subiteration in an implicit subpatch and one explicit solve in an explicit subpatch are approximately equal. Figure 12 shows a two-dimensional slice of a portion of the domain with this discretization.

The boundary conditions at the surface of the sphere are no-slip and isothermal ($T = 1$), and an absorbing layer is used at the outer edges of the domain (as described in the previous section) using the parameters $A_a = 5.0$, $p_a = 4$, and $w_a = 1.0$ in equation (34). As in Section 10.1, we use the method of manufactured initial conditions: the fields are initialized with the functions

$$\rho = T = 1, \quad v = w = 0, \quad u = u_0 \psi_s(r - r_s), \tag{40}$$

where u_0 is the freestream velocity and ψ_s is the smooth step function (36). A right-hand side source term is added to the equations that enforce the initial condition, which is made to vanish smoothly by multiplying it with $\psi_s(1 - t/t_0)$.

Figure 13 shows a slice of the streamlines in the x-z plane at $t = 12$. At this early time in the simulation, the developing flow is still laminar and axially symmetric (about the x-axis), exhibiting the characteristic axially symmetric vortex behind the sphere. Figure 14 shows a slice of the x-velocity, showing vortices developing around areas of low density relative to the freestream density. In Figure 15, snapshots of density isosurfaces indicate the shedding of "hairpin" shaped vortices, the character of which are qualitatively consistent with the results of similar experiments and numerical simulations for incompressible flow [46, 47, 48, 49].

11. Conclusions

We presented a new class of solvers for the subsonic compressible Navier-Stokes equations in general two- and three-dimensional domains. The proposed hybrid implicit-explicit methodology maximizes efficiency and accuracy by incorporating novel linear-cost implicit time-marching solvers based on higher-order backward differentiation formulae (BDF) and the alternating direction implicit approach (ADI), efficient explicit solvers based on Adams-Bashforth schemes of up to fourth order, essentially dispersionless spectral spatial discretizations, and an overset domain decomposition strategy that balances the cost and interactions among regions that are evolved implicitly and explicitly. As shown in [17], the implicit methodology is quasi-unconditionally stable—it is not subject to CFL constraints for adequately resolved flows (Section 4.2.1)—and it can deliver orders of time accuracy between two and six even in the presence of general boundary conditions. The proposed multi-domain parallel implicit-explicit implementations exhibit high-order convergence in space and time, robust stability properties, limited dispersion, and high parallel efficiency.

Acknowledgements

The authors gratefully acknowledge support by NSF and AFOSR and DARPA through contracts DMS-1411876, DMS-1714169, FA9550-15-1-0043 and HR00111720035, and the NSSEFF Vannevar Bush Fellowship under contract number N00014-16-1-2808.

References

[1] H. A. Schwarz, Ueber einige Abbildungsaufgaben., Journal für die reine und angewandte Mathematik 70 (1869) 105–120.
[2] E. A. Volkov, The method of composite meshes for finite and infinite regions with piecewise smooth boundary, Trudy Matematicheskogo Instituta im. VA Steklova 96 (1968) 117–148.
[3] G. Starius, Composite mesh difference methods for elliptic boundary value problems, Numerische Mathematik 28 (1977) 243–258.
[4] J. L. Steger, F. C. Dougherty, J. A. Benek, A chimera grid scheme, Advances in Grid Generation (1983).
[5] G. Chesshire, W. D. Henshaw, Composite overlapping meshes for the solution of partial differential equations, Journal of Computational Physics 90 (1990) 1–64.
[6] D. L. Brown, W. D. Henshaw, D. J. Quinlan, Overture: Object-oriented tools for overset grid applications, AIAA paper No. 99 9130 (1999).
[7] M. R. Visbal, D. V. Gaitonde, On the use of higher-order finite-difference schemes on curvilinear and deforming meshes, Journal of Computational Physics 181 (2002) 155–185.
[8] S. E. Sherer, J. N. Scott, High-order compact finite-difference methods on general overset grids, J. Comput. Phys. 210 (2005) 459–496.
[9] R. M. Beam, R. F. Warming, An implicit finite-difference algorithm for hyperbolic systems in conservation-law form, Journal of Computational Physics 22 (1976) 87–110.
[10] R. M. Beam, R. Warming, An implicit factored scheme for the compressible Navier-Stokes equations, AIAA journal 16 (1978) 393–402.
[11] O. P. Bruno, M. Cubillos, Higher-order in time "quasi-unconditionally stable" ADI solvers for the compressible Navier–Stokes equations in 2d and 3d curvilinear domains, Journal of Computational Physics 307 (2016) 476–495.
[12] N. Albin, O. P. Bruno, A spectral FC solver for the compressible Navier–Stokes equations in general domains I: Explicit time-stepping, Journal of Computational Physics 230 (2011) 6248–6270.
[13] F. Amlani, O. P. Bruno, An FC-based spectral solver for elastodynamic problems in general three-dimensional domains, J. Comput. Phys. 307 (2016) 333–354.
[14] O. P. Bruno, M. Lyon, High-order unconditionally stable FC-AD solvers for general smooth domains I. Basic elements, Journal of Computational Physics 229 (2010) 2009–2033.
[15] D. W. Peaceman, H. H. Rachford, Jr., The numerical solution of parabolic and elliptic differential equations, Journal of the Society for Industrial and Applied Mathematics 3 (1955) 28–41.
[16] O. P. Bruno, E. Jimenez, Higher-order linear-time unconditionally stable alternating direction implicit methods for nonlinear convection-diffusion partial differential equation systems, Journal of Fluids Engineering 136 (2014) 060904–060904.
[17] O. P. Bruno, M. Cubillos, On the Quasi-unconditional Stability of BDF-ADI Solvers for the Compressible Navier–Stokes Equations and Related Linear Problems, SIAM Journal on Numerical Analysis 55 (2017) 892–922.
[18] H. Bijl, M. H. Carpenter, V. N. Vatsa, C. A. Kennedy, Implicit time integration schemes for the unsteady compressible Navier–Stokes equations: laminar flow, Journal of Computational Physics 179 (2002) 313–329.
[19] U. M. Ascher, S. J. Ruuth, B. T. Wetton, Implicit-explicit methods for time-dependent partial differential equations, SIAM Journal on Numerical Analysis 32 (1995) 797–823.
[20] T. J. Elling, GPU-accelerated Fourier-continuation solvers and physically exact computational boundary conditions for wave scattering problems, PhD, California Institute of Technology, 2013. URL: `http://resolver.caltech.edu/CaltechTHESIS:07092012-144406693`.
[21] S. K. Lele, Compact finite difference schemes with spectral-like resolution, Journal of Computational Physics 103 (1992) 16–42.
[22] G. Ashcroft, X. Zhang, Optimized prefactored compact schemes, Journal of Computational Physics 190 (2003) 459–477.
[23] K. Mattsson, M. Svärd, M. Shoeybi, Stable and accurate schemes for the compressible Navier–Stokes equations, Journal of Computational Physics 227 (2008) 2293–2316.
[24] J. Nordström, J. Gong, E. Van der Weide, M. Svärd, A stable and conservative high order multi-block method for the compressible Navier–Stokes equations, Journal of Computational Physics 228 (2009) 9020–9035.

[25] M. Svärd, M. H. Carpenter, J. Nordström, A stable high-order finite difference scheme for the compressible Navier–Stokes equations, far-field boundary conditions, Journal of Computational Physics 225 (2007) 1020–1038.

[26] M. Svärd, J. Nordström, A stable high-order finite difference scheme for the compressible Navier–Stokes equations: No-slip wall boundary conditions, Journal of Computational Physics 227 (2008) 4805–4824.

[27] C. K. W. Tam, K. A. Kurbatskii, Multi-size-mesh multi-time-step dispersion-relation-preserving scheme for multiple-scales aeroacoustics problems, International Journal of Computational Fluid Dynamics 17 (2003) 119–132.

[28] C. K. W. Tam, J. C. Webb, Dispersion-relation-preserving finite difference schemes for computational acoustics, Journal of Computational Physics 107 (1993) 262–281.

[29] J. P. Boyd, Chebyshev and Fourier spectral methods, Courier Dover Publications, 2001. URL: http://books.google.com/books?hl=en&lr=&id=lEWnQWyzLQYC&oi=fnd&pg=PA1&dq=john+boyd+spectral+methods&ots=WSaKAHi4ux&sig=g_sNftRiP7c7TVPmqKB8GTdfrgU.

[30] J. S. Hesthaven, S. Gottlieb, D. Gottlieb, Spectral methods for time-dependent problems, volume 21, Cambridge University Press, 2007. URL: http://books.google.com/books?hl=en&lr=&id=dpZg1YEr4GEC&oi=fnd&pg=PA252&dq=hesthaven+spectral+methods&ots=Aj86AnHSad&sig=tN2RiiBj7V-XNBx7kg5e3rKfybQ.

[31] C. Canuto, M. Y. Hussaini, A. Quarteroni, T. A. Zang, Spectral methods: fundamentals in single domains, Springer Science & Business Media, 2007.

[32] J. D. Lambert, Numerical methods for ordinary differential systems: the initial value problem, John Wiley & Sons, Inc., 1991. URL: http://dl.acm.org/citation.cfm?id=129839.

[33] F. M. White, Viscous fluid flow, McGraw-Hill New York, 2006. URL: http://www.ulb.tu-darmstadt.de/tocs/21266050.pdf.

[34] O. P. Bruno, Y. Han, M. M. Pohlman, Accurate, high-order representation of complex three-dimensional surfaces via Fourier continuation analysis, Journal of Computational Physics 227 (2007) 1094–1125.

[35] M. Lyon, O. P. Bruno, High-order unconditionally stable FC-AD solvers for general smooth domains II. Elliptic, parabolic and hyperbolic PDEs; theoretical considerations, Journal of Computational Physics 229 (2010) 3358–3381.

[36] S. Abarbanel, D. Gottlieb, M. H. Carpenter, On the removal of boundary errors caused by Runge-Kutta integration of nonlinear partial differential equations, SIAM Journal on Scientific Computing 17 (1996) 777–782.

[37] M. H. Carpenter, D. Gottlieb, S. Abarbanel, W.-S. Don, The theoretical accuracy of Runge-Kutta time discretizations for the initial boundary value problem: a study of the boundary error, SIAM Journal on Scientific Computing 16 (1995) 1241–1252.

[38] D. V. Gaitonde, M. R. Visbal, Pade-type higher-order boundary filters for the Navier-Stokes equations, AIAA Journal 38 (2000) 2103–2112.

[39] M. Cubillos, General-domain compressible Navier-Stokes solvers exhibiting quasi-unconditional stability and high-order accuracy in space and time, PhD, California Institute of Technology, 2015. URL: http://resolver.caltech.edu/CaltechTHESIS:05082015-184801592.

[40] A. Bondeson, T. Rylander, P. Ingelström, Computational Electromagnetics, Texts in Applied Mathematics, Springer New York, 2005. URL: https://books.google.com/books?id=PofGtToqtN0C.

[41] L. Trefethen, Group velocity in finite difference schemes, SIAM Review 24 (1982) 113–136.

[42] Y. Saad, M. H. Schultz, GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems, SIAM Journal on Scientific and Statistical Computing 7 (1986) 856–869.

[43] O. P. Bruno, A. Prieto, Spatially dispersionless, unconditionally stable FC–AD solvers for variable-coefficient PDEs, Journal of Scientific Computing 58 (2014) 331–366.

[44] S. Völkner, J. Brunswig, T. Rung, Analysis of non-conservative interpolation techniques in overset grid finite-volume methods, Computers & Fluids 148 (2017) 39–55.

[45] A. Kageyama, T. Sato, "Yin-Yang grid": An overset grid in spherical geometry, Geochemistry, Geophysics, Geosystems 5 (2004) Q09005.

[46] D. Kim, H. Choi, Laminar flow past a sphere rotating in the streamwise direction, Journal of Fluid Mechanics 461 (2002) 365–386.

[47] T. A. Johnson, V. C. Patel, Flow past a sphere up to a Reynolds number of 300, Journal of Fluid Mechanics 378 (1999) 19–70.

[48] M. Kiya, H. Ishikawa, H. Sakamoto, Near-wake instabilities and vortex structures of three-dimensional bluff bodies: a review, Journal of Wind Engineering and Industrial Aerodynamics 89 (2001) 1219–1232.

[49] H. Sakamoto, H. Haniu, A study on vortex shedding from spheres in a uniform flow, Journal of Fluids Engineering 112 (1990) 386–392.